THE LINUX FOUNDATION | Research

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Secure Software Development Education 2024 Survey

Understanding Current Needs

Marco Gerosa, Ph.D., *Northern Arizona University*
David A. Wheeler, Ph.D., *The Linux Foundation*
Stephen Hendrick, *The Linux Foundation*

Foreword by
Christopher Robinson, *Intel*
Dave Russo, *Red Hat*

June 2024

Secure Software Development Education 2024 Survey

28% of professionals directly involved in software development are **not familiar** with secure software development.

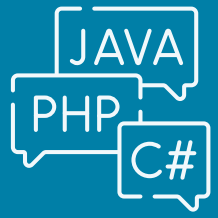
Software developers with **less than one year of experience** report the highest lack of familiarity (75%)

69% of professionals rely on on-the-job experience as a learning resource for secure software development, but **it can take more than 5 years of such experience** to achieve familiarity.

50% of professionals identify **a lack of training as a major challenge** for implementing secure software development, with this issue being particularly pronounced among data science roles (73%).

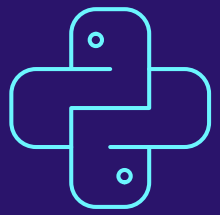
53% of professionals, especially those in system operations (72%), have not taken a course on secure software development, largely due to **the lack of awareness about good courses** (44%).

79% of professionals consider language-agnostic courses highly important, compared with 54% who attribute the same level of importance to language-specific courses.

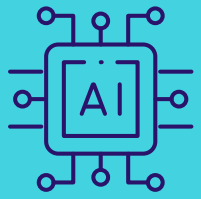


Popular language-agnostic courses include **security architecture** (64%), **security education and guidance** (64%), and **secure implementation** (63%).

Training needs vary significantly based on **professional roles and experience levels**.

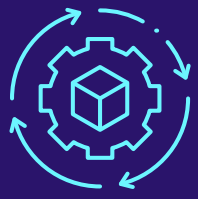


Python is highly favored for language-specific training, with 71% of respondents expressing a preference, although C and Java are selected more frequently when respondents rank their top choices.



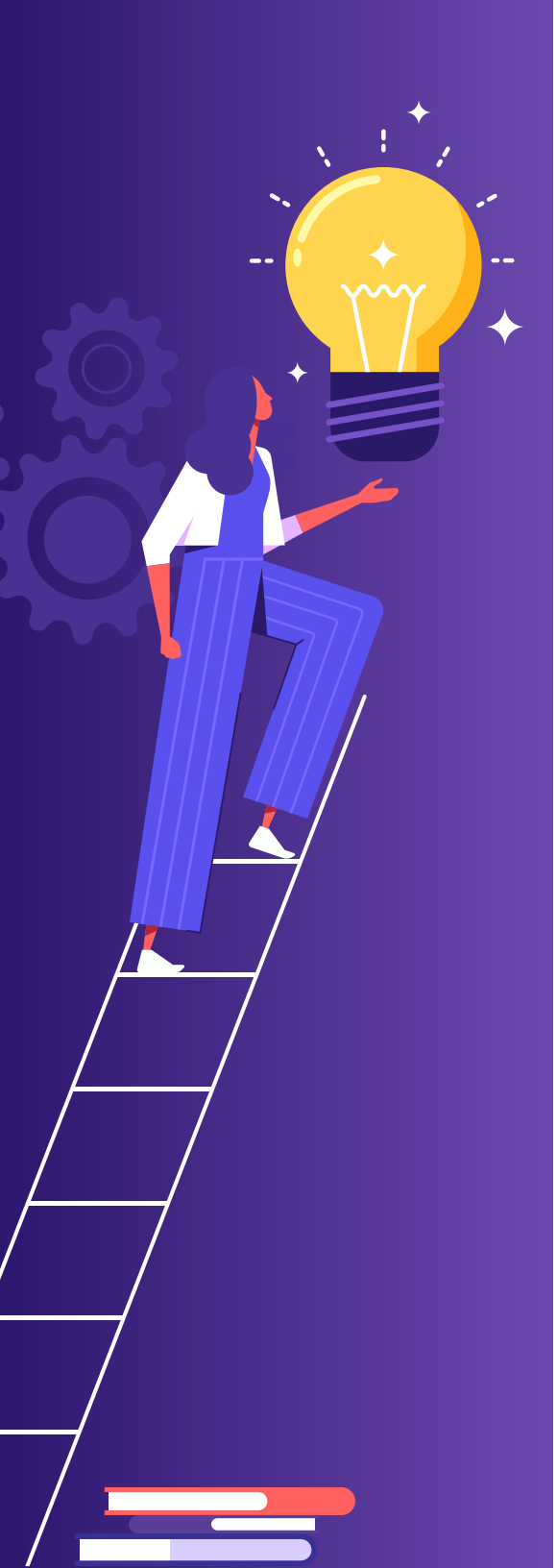
57% of respondents identify **AI and ML security** as a critical area for future innovation and attention in secure software development.

56% of respondents see **supply chain security** as a crucial area needing increased focus and innovation.



To start mitigating the need for more secure software development education, the OpenSSF selected **Security Architecture as the topic of a new course**.

Contents



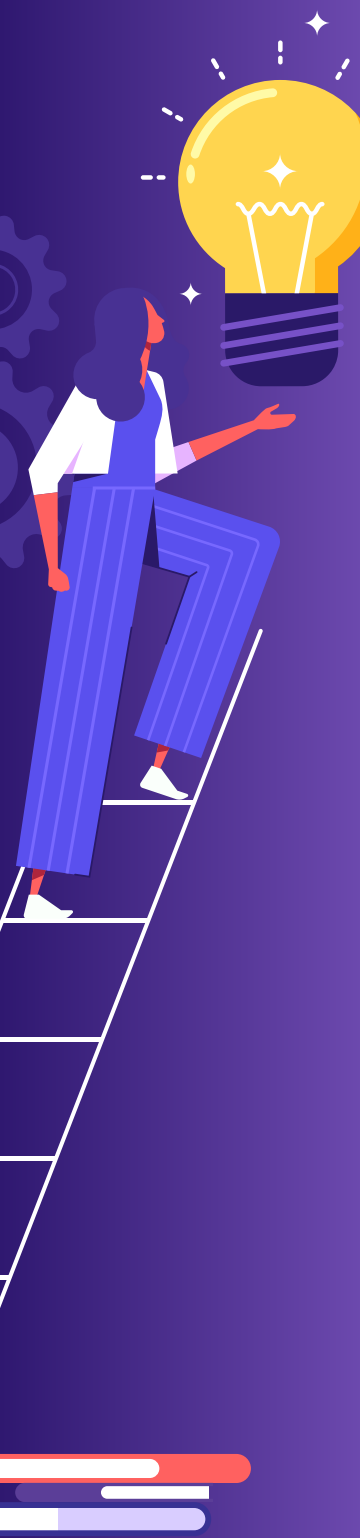
Foreword

Above all else, education is a tool that, once obtained, is always available to the developer no matter what language, IDE, or scanner they may be working in or have access to. I am pleased to have participated in this Secure Software Development Education survey and that we now can share the results of the Linux Foundation's (LF) research with the community. We've already started reacting to some of the initial findings, and now that the full report is available, I look forward to helping empower developers of all skill levels, experiences, and backgrounds based on the important feedback that the community has provided.

Christopher Robinson, *Intel*, Co-Chair of the OpenSSF Education Special Interest Group and Chair of the OpenSSF Technical Advisory Council

No matter how sophisticated developer tools become, the knowledge and mindset of the individuals designing and writing the code have the biggest impact on its overall quality, especially when it comes to developing securely. Understanding what developers need to know and effectively delivering that information to them in a digestible way is key to enabling them to keep secure practices top of mind and to be able to effectively implement them. The results of the OpenSSF Secure Software Development Education survey reinforce the need for these educational materials and the benefits they will provide. Our group will leverage this information to improve and expand the overall availability and quality of this training to the open source community and encourage others to do the same.

Dave Russo, *Red Hat*, Co-Chair of the OpenSSF Education Special Interest Group



Chapter 1: Introduction

Ensuring software security has never been more critical. Software vulnerabilities can lead to catastrophic consequences in many areas, from financial transactions and healthcare management to national security and everyday communication. A data breach in the U.S. costs $9.44 million on average per incident, according to an IBM Report (2023)[1]. A Verizon report (2021)[2] shows that 43% of all breaches are linked to software vulnerabilities due to poor software development practices.

The evolution of cyberthreats has shown that attackers continuously find ways to exploit software weaknesses. By prioritizing secure coding practices, regular security assessments, and proactive threat modeling, developers can build resilient systems that protect sensitive data and ensure user trust. Secure software development is not merely an additional layer in the software development process but an integral aspect of it.

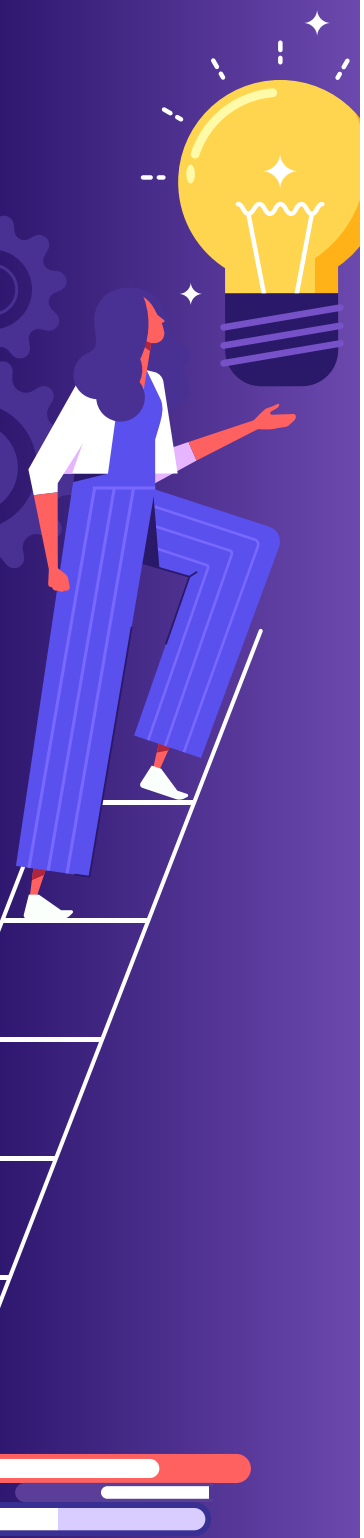
Despite its critical importance, many developers lack the necessary knowledge and skills to implement secure software development effectively. Many educational programs focus primarily on functionality and efficiency, often neglecting security training. The historical emphasis on functionality over security has been a pervasive issue in software development. This focus can be traced back to the early days of computing, where the primary goal was to create functional and reliable systems to perform specific tasks. Security was often an afterthought, if it was considered at all.

Despite its critical importance, many developers lack the necessary knowledge and skills to implement secure software development effectively. Many educational programs focus primarily on functionality and efficiency, often neglecting security training.

The first step in addressing secure software development is recognizing the existing knowledge gap and identifying priority areas to create additional training. With this goal in mind, the Open Source Security Foundation (OpenSSF) and Linux Foundation (LF) Research partnered to conduct a worldwide survey of software development professionals to assess their secure software development education needs. This research seeks to promote a “security by design” approach to software developer education and to enhance security education programs.

1 **https://www.ibm.com/reports/data-breach**

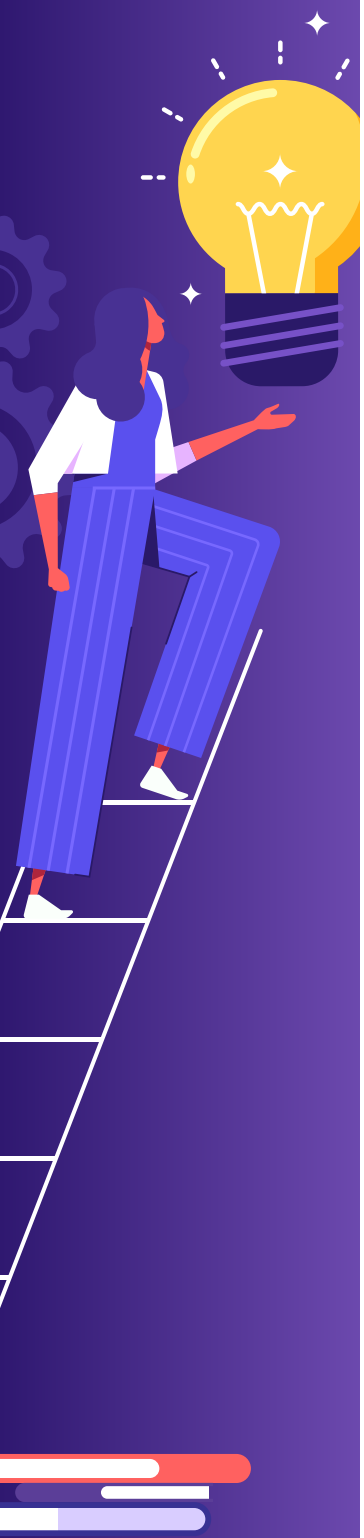
2 **https://www.verizon.com/about/news/verizon-2021-data-breach-investigations-report**



From March 1 to April 29, 2024, the survey received 398 valid responses from professionals involved with software development, which are the basis for the analysis presented in this report. The survey included questions on demographics, experience, and current perspectives for secure software development, along with survey questions focusing on educational needs for secure software development. For more information about the research approach and demographics, see Chapter 5.

Our key findings are as follows:

1. A large portion (28%) of professionals directly involved in software development and deployment, including system operations, software developers, committers, and maintainers, report not being familiar with secure software development.
2. Software developers with less than one year of experience report the highest lack of familiarity (75%).
3. 69% of professionals rely on on-the-job experience as a main learning resource, but it takes at least five years of such experience to achieve a minimum level of familiarity.
4. Lack of training is a major challenge for many professionals (50%), particularly those in data science roles (73%).
5. Most professionals (53%), especially system operations professionals (72%), have not taken a course on the topic, especially because they are not aware of a good course (44%).
6. Most professionals (79%) deem language-agnostic courses highly important for secure software development, overshadowing the 54% who view language-specific courses as highly important.
7. Popular language-agnostic courses include security architecture, security education and guidance, and secure implementation.
8. Training needs vary significantly based on professional roles and experience levels, evidencing the need for diverse educational offerings in secure software practices.
9. Python is highly favored for language-specific training, with 71% of respondents expressing a preference, although when ranking their top choices, C and Java were selected more frequently, suggesting a nuanced demand for programming language education.
10. Emerging security concerns such as AI and ML security and supply chain are seen as critical future areas for innovation and attention, identified by 57% and 56% of respondents, respectively.
11. Based on these findings, the OpenSSF has decided to create a new course on security architecture, as explained in Chapter 4. Although we've selectively highlighted several key findings here, we've made all the data openly available for you to explore.

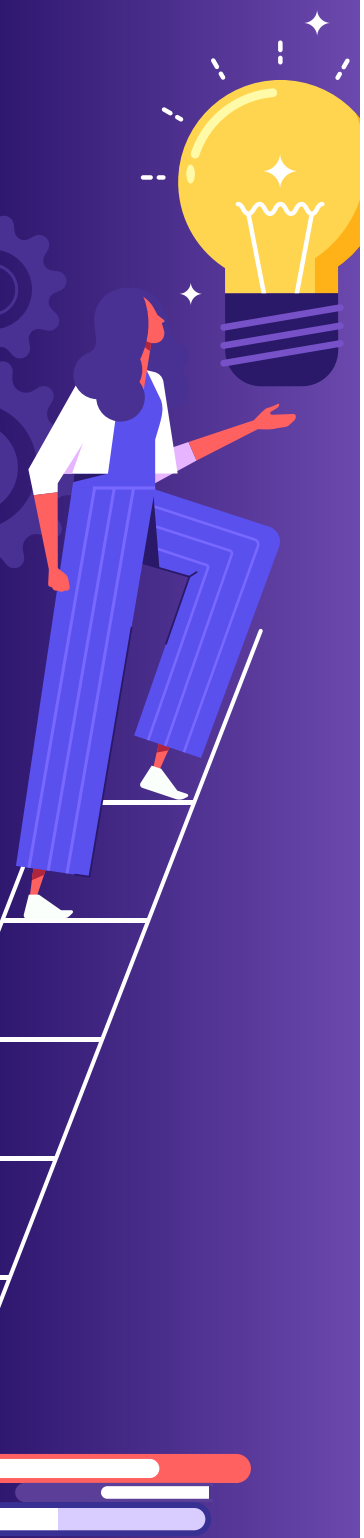


Chapter 2: The need for more training

This chapter explores the need for more training. We assess the respondents' familiarity with secure development practices, the challenges in implementing these practices, and which learning resources professionals utilize. This analysis establishes a foundation for understanding the need for additional training in this area.

The main findings of this chapter are as follows:

1. Many professionals (28%) involved with software development are not familiar with secure software development.
2. Key roles in software development and deployment, such as system operations (39%) and software developers (27%), and in open source software (OSS) in particular, such as open source program office (OSPO) members (38%), committers (29%), and maintainers (23%), have a high number of professionals not familiar with the topic.
3. Even some security team members (16%) are not familiar with the topic.
4. Being experienced with software development does not imply familiarity with the topic, with at least 20% not being familiar regardless of the number of years of experience.
5. At least five years of practical experience in the topic are necessary for at least 90% of professionals to consider themselves familiar with it.
6. Among the professionals who have not taken a course on secure software development, very few (13%) said that it was because they feel they already know enough about the topic.
7. Lack of awareness and training is the second most common challenge in implementing secure software development capabilities within organizations (50%), only behind lack of time (58%).
8. Lack of awareness and training is particularly challenging for 73% of data science professionals.
9. Informal methods such as self-study (74%) and on-the-job experience (69%) are the primary learning resources for the topic.
10. The majority (60%) of security team members have taken a course on the topic, while a minority of other key roles, such as software developers (48%) and system operations professionals (28%), have taken one.
11. The top reason, reported by 44% of the respondents, for not taking a course is the lack of knowledge about a good course on the topic.
12. Few professionals (up to 13%) report that they do not need a course on the topic.
13. Only 25% of organizations report using OpenSSF education materials, and the top reason is a lack of awareness.
14. Most respondents (74%) prefer self-paced training materials.



Many professionals are not familiar with secure software development

Nearly one-third of all software development professionals do not feel familiar with secure software development, as observed in Figure 1. There is also a chance that even those who report familiarity with secure software development do not know how to apply it in practice. These results are corroborated by Figure 2, which shows that only 13% of the respondents see themselves as not needing training because they already know enough about the subject. Worryingly, as observed in Figure 1, professionals in some critical roles in the development process lack familiarity with secure software development practices.

For those whose primary role is software development, it is concerning that 27% report being unfamiliar with secure software development practices. This fact is particularly troubling given that software developers are at the forefront of creating and maintaining the code that runs a company's applications and systems. The lack of familiarity in over one-quarter of developers indicates a significant gap in essential knowledge that could lead to the introduction of security vulnerabilities during the development process. For companies, this emphasizes the urgent need to integrate comprehensive security training into the standard developer curriculum and ensure that secure coding practices are a foundational element of the software development lifecycle.

If anything, the data suggests that things are even worse than they first appear, once other reports are considered. In a 2022 study by Secure Code Warrior, 89% of responding developers claimed that they'd received "sufficient" training in secure coding skills, yet when they were asked about specifics, the majority of the respondents admitted that they were not familiar with common software vulnerabilities, how to avoid them, and how they can be exploited. Indeed, 86% of the developers in that study stated that they found it challenging to practice secure coding, an odd result if they'd really received sufficient training.[3] A plausible explanation is that knowledge of how to develop secure software is so rare that developers overestimate their knowledge, presuming a familiarity that isn't justified. This shouldn't be surprising, as even the developers who go to university are unlikely to learn how to do it. A 2021 study pointed out that of U.S. News's top 24 computer science schools, only one—University of California San Diego—requires undergraduates to learn about security.[4] In short, our data is likely to show the situation in a more positive light than reality, due to the widespread lack of understanding of even the basics.

The data also reveals that system operations and OSPO team members report the highest levels of unfamiliarity with secure software development (39% and 38%, respectively). This fact is concerning, as these roles are critical in managing and maintaining software infrastructure and open source initiatives, both of which are fundamental to a company's overall security posture. Security team members reported the lowest level of unfamiliarity at 16%. Even though this indicates that those specifically tasked with security are more knowledgeable about secure development practices, it is worrisome that not all these professionals consider themselves at least familiar with the area. For companies, these results highlight the need for cross-departmental training programs and initiatives to foster a culture of secure software development awareness.

3 The State of Developer-Driven Security Survey, Secure Code Warrior, 2022, **https://discover.securecodewarrior.com/state-of-developer-driven-security-2022.html**

4 **https://www.appsecengineer.com/blog/developer-security-at-universities**

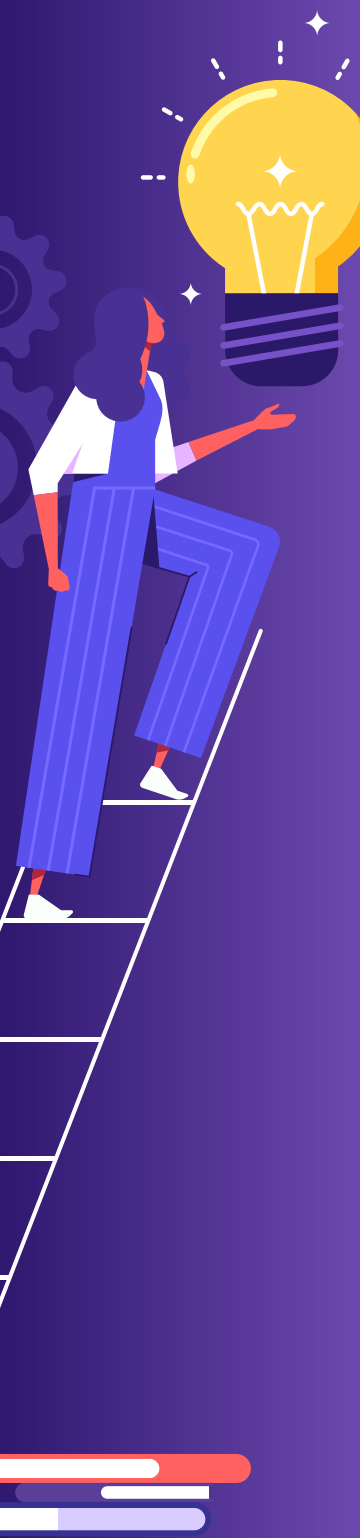
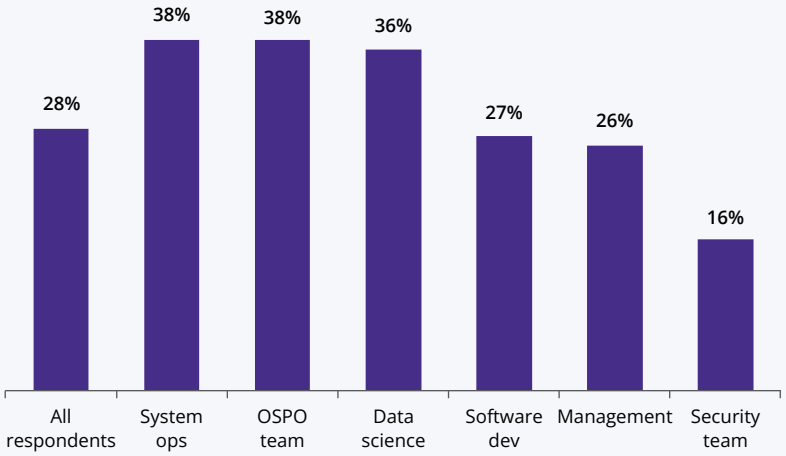


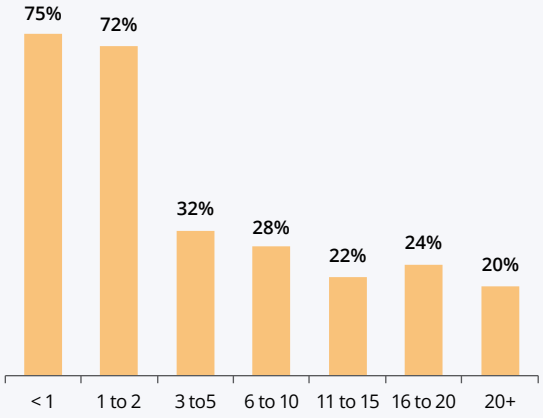
FIGURE 1

PERCENTAGE OF RESPONDENTS NOT FAMILIAR WITH SECURE SOFTWARE DEVELOPMENT

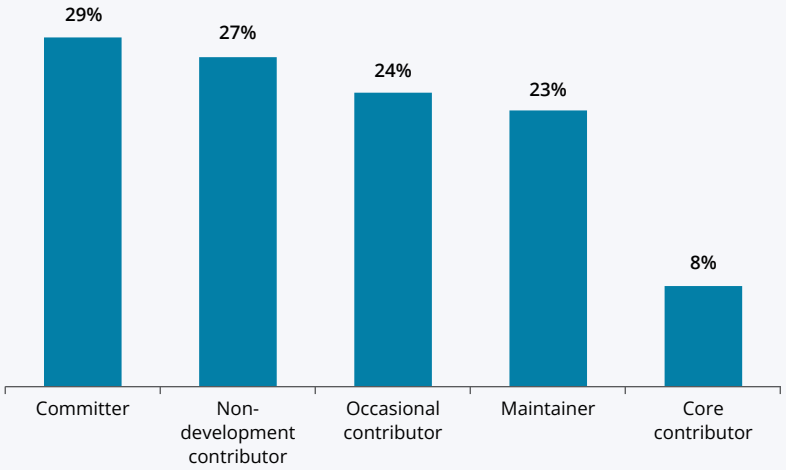
Segmented by professional role



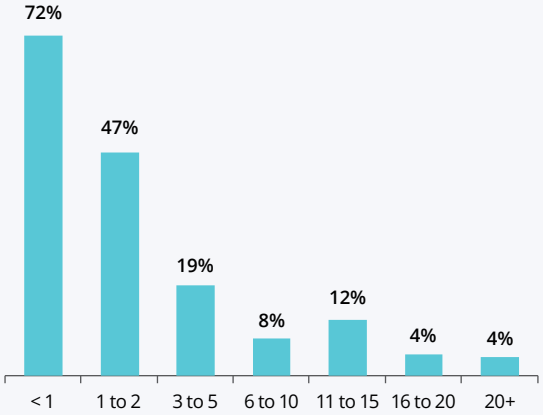
Segmented by years of experience in software development



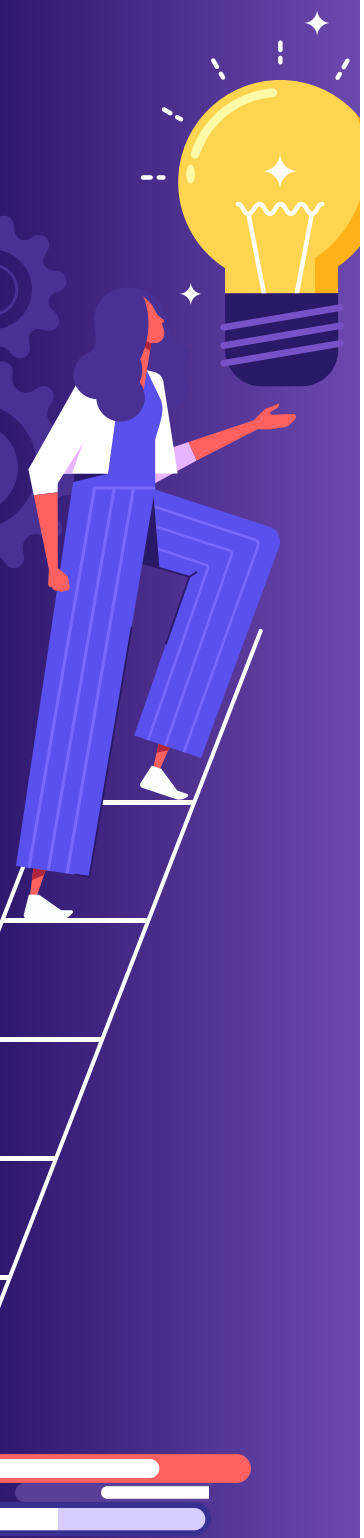
Segmented by open source software role



Segmented by years of experience in secure software development



2024 SecEd Survey, Q14 by Q5, Q8, Q15, Q16, Sample Size = 396, Low familiarity represents those who answered “Not familiar at all” or “Somewhat familiar”

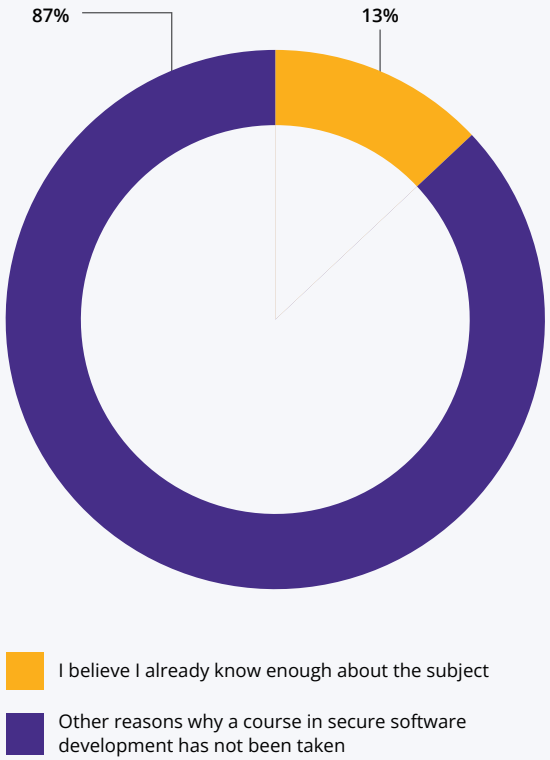


Narrowing the results for those who contribute to OSS, we can see that more than one-quarter of committers and maintainers do not consider themselves familiar with secure software development. This fact suggests that many developers who write and send code directly to open source repositories and review others' work are not familiar with the area. Given that OSS comprises most of the scaffolding technologies that many modern systems are built upon, the lack of security knowledge can bring generalized threats, as we observe from time to time.

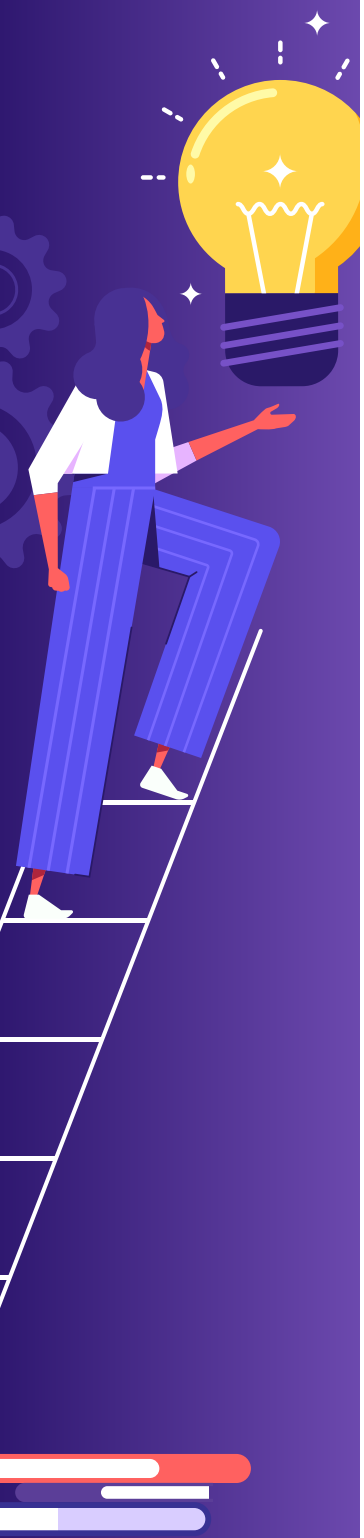
The survey also highlights a stark difference in familiarity based on years of experience. Software developers with less than one year of experience report the highest lack of familiarity at 75%, with this number dropping to 72% for those with one to two years of experience. Similarly, 72% of those with less than one year of specific experience in secure software development report a lack of familiarity, while this number drops to 47% for those with one to two years of experience. Despite these numbers declining with increased experience, 20% of professionals with more than 20 years of general experience still report a lack of familiarity with the field. This indicates that even highly experienced developers may not necessarily be knowledgeable about secure software development, and it often takes many years of specific practical experience to gain familiarity. For companies, this highlights the importance of incorporating secure software development training early in a software professional's career. It also suggests that companies should invest in onboarding programs that emphasize secure coding practices and provide continuous education opportunities to bridge this knowledge gap.

FIGURE 2

PERCENTAGE OF RESPONDENTS WHO REPORT NOT HAVING TAKEN A COURSE ON SECURE SOFTWARE DEVELOPMENT BECAUSE THEY BELIEVE THEY ALREADY HAVE ENOUGH KNOWLEDGE ON THE TOPIC



2024 SecEd Survey, Q31, Sample Size = 150



The need for awareness and training is a major challenge for secure software development

Effectively implementing secure software development and deployment brings many challenges. Our results indicate that the need for security awareness and training is one of the top challenges for organizations. With half of the respondents reporting this challenge, it ranks only below time constraints, as depicted in Figure 3. Since time constraints are a common problem across many organizations, to successfully address these needs, most organizations will need to address security awareness and training with systematic and structured education programs. These programs should be integrated into the organizational culture and workflows, ensuring that they are regular and mandatory. Additionally, fostering a culture of continuous learning and security-minded thinking across all departments can enhance the effectiveness of these educational efforts.

The perception of a need for more security awareness and training as a challenge for implementing secure software development and deployment varies depending on the professional role, as pointed out in Figure 4. Data science roles report the highest level of concern, with 73% of respondents

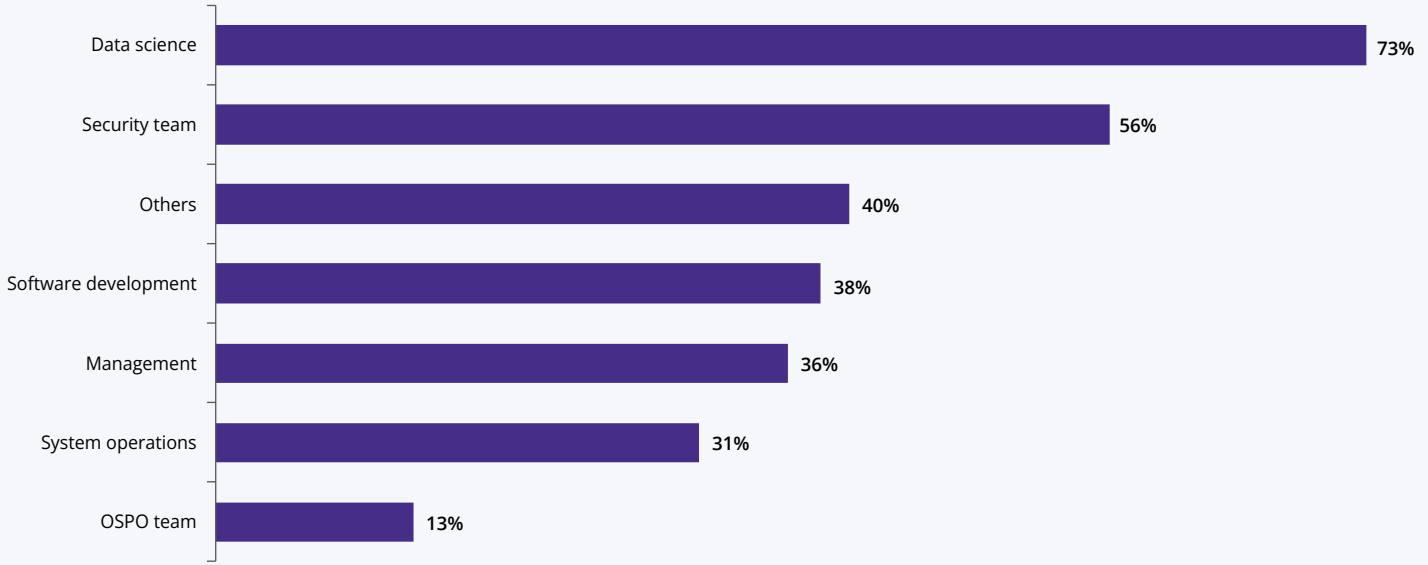
FIGURE 3

BIGGEST CHALLENGES IN IMPLEMENTING SECURE SOFTWARE DEVELOPMENT AND DEPLOYMENT IN AN ORGANIZATION

2024 SecEd Survey, Q28, Sample Size = 324, Total Mentions = 1,224

FIGURE 4

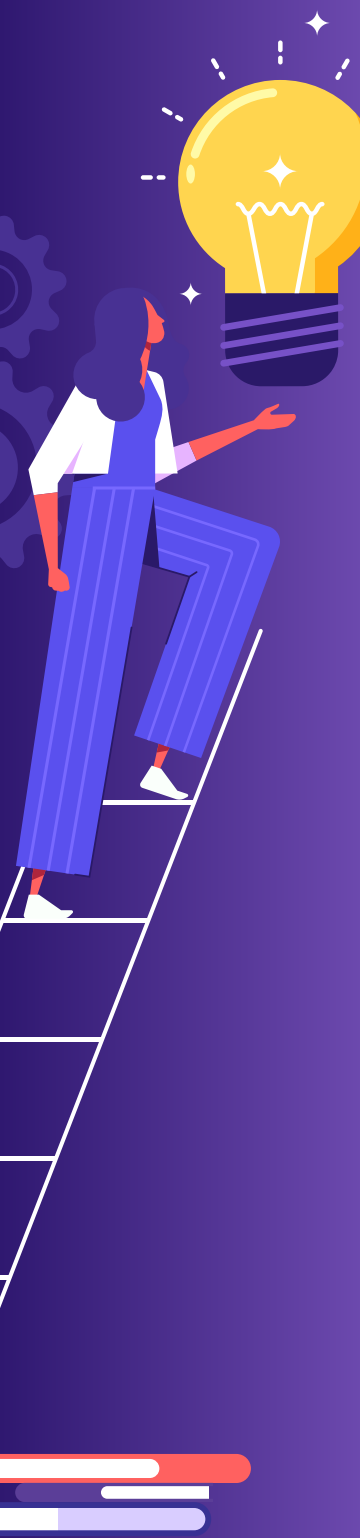
PERCENTAGE OF RESPONDENTS WHO REPORTED LACK OF AWARENESS AND TRAINING AS A CHALLENGE FOR IMPLEMENTING SECURE SOFTWARE DEVELOPMENT AND DEPLOYMENT, SEGMENTED BY RESPONDENT ROLE



2024 SecEd Survey, Q28 by Q5, Sample Size = 398

indicating that this is a significant challenge. This high percentage likely reflects the fact that data science professionals often come from academic areas not well versed in software engineering practices, including secure coding standards and methodologies. This gap in their training is worrisome, since data scientists increasingly deploy models and algorithms directly into production environments, and the lack of security practices can lead to vulnerabilities and expose large volumes of sensitive data. This is especially the case in the use of machine learning (ML) systems, where data may be used to train models that are directly deployed in production systems. This result emphasizes the need for more robust and specific training in data protection.

Security team members also feel that the need for security awareness and training is a concern for implementing secure software development, with 56% of professionals reporting this challenge. This high percentage reflects the security team's unique perspective on the organization's overall preparedness. As the primary protectors against cyberthreats, they are acutely aware of the discrepancies between ideal security



protocols and the actual practices adopted by software developers, leading to gaps in the organization's security infrastructure. This gap underscores the need for more organization-wide secure software development education to prevent vulnerabilities due to better security awareness across the development lifecycle. The need for more awareness and training is also shared with many professionals in other roles, including software development (38%), management (36%), and system operations (31%).

A large number of respondents have not taken any courses on secure software development

Many software development professionals still favor informal methods over university educational courses. Figure 5 demonstrates that the prevalent method for learning secure software development is self-study, with 74% of respondents reporting utilizing resources such as online tutorials, videos, and books as their main learning method. This method is closely followed by 69% who credit accumulated on-the-job experience. These popular methods have their drawbacks. Self-study relies heavily on individual initiative and often lacks the comprehensive curriculum and expert guidance found in educational courses, which can lead to gaps in knowledge. On-the-job learning, while practical, can also be inconsistent, depending heavily on the locally available expertise, specific projects, and security challenges encountered in the workplace. Moreover, errors common among those learning can inadvertently be incorporated into production code, compromising system security.

A course on secure software development can equip professionals with the skills and knowledge to identify, mitigate, and prevent security vulnerabilities in software, thereby enhancing product security and protecting their organizations from potential cyberthreats. However, our findings indicate that many professionals in the field have not yet taken such a course. As observed in Figure 6, 47% of the respondents reported having taken a course on secure software development. Among specific groups, the security team leads with 60% participation, confirming their central role in cybersecurity initiatives. For most other roles, the percentages range from 44% to 50%, which means that most professionals in these roles have not had such training. The small percentage refers to system operation professionals at 28%. In many modern IT environments, system operation personnel increasingly write software as part of their jobs, giving rise to the DevOps phenomenon. Neglecting security practices in these applications can introduce security vulnerabilities and compromise the whole ecosystem. Training these professionals is also essential because this knowledge enables them to work more collaboratively with software development teams to ensure that security considerations are integrated throughout the lifecycle of the systems they support, enhancing overall organizational security.

Notably, fewer than one-quarter of the respondents have learned about secure software development through formal academic courses (e.g., in colleges and universities), as depicted in Figure 5. This low percentage suggests that skill gaps originate from academic settings and need to be addressed through additional training before professionals are onboarded onto software development projects. As noted earlier, this is likely simply because it's not required in most undergraduate settings. This additional training ensures that professionals are adequately prepared for the security demands of modern software development.

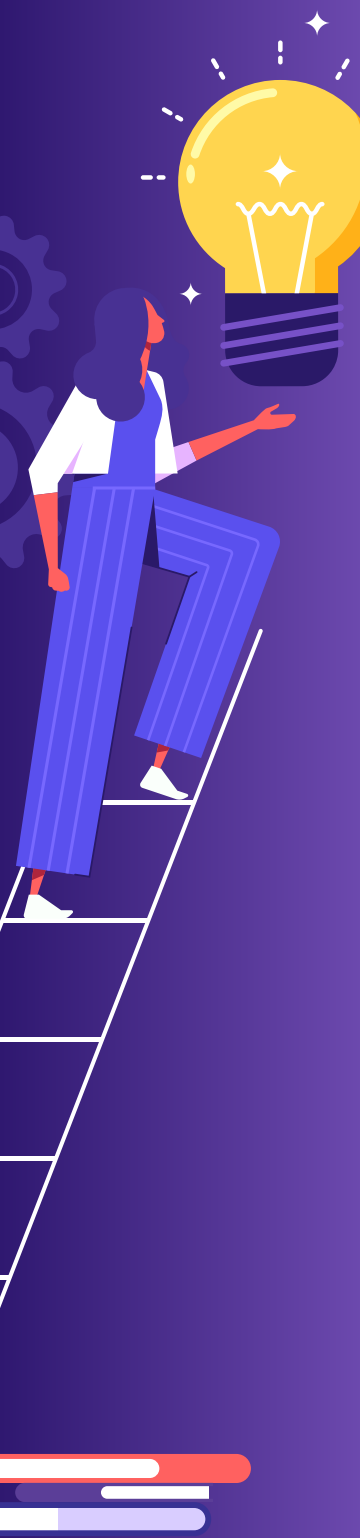


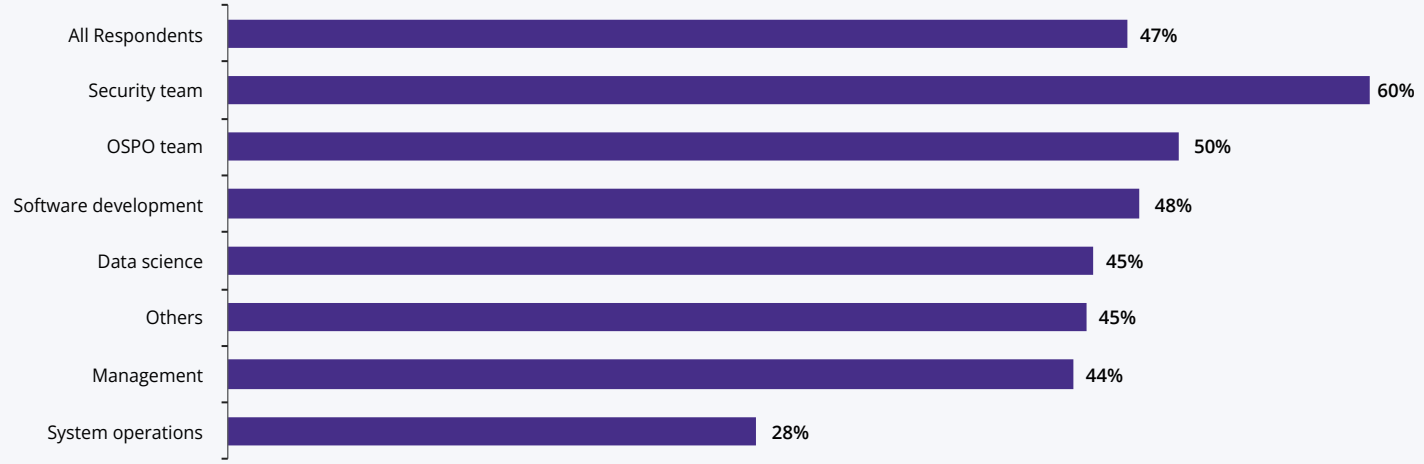
FIGURE 5

PRIMARY LEARNING RESOURCES FOR SECURE SOFTWARE DEVELOPMENT

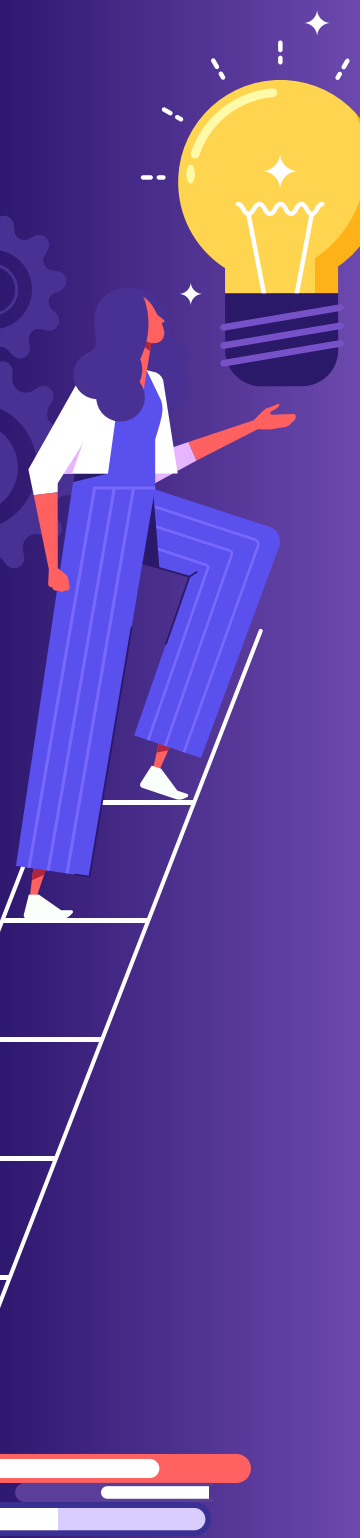
2024 SecEd Survey, Q17, Sample Size = 398, Total Mentions = 948

FIGURE 6

PERCENTAGE OF RESPONDENTS WHO HAVE TAKEN A COURSE IN SECURE SOFTWARE DEVELOPMENT



2024 SecEd Survey, Q20 by Q5, Sample Size = 383, DKNS excluded from the analysis



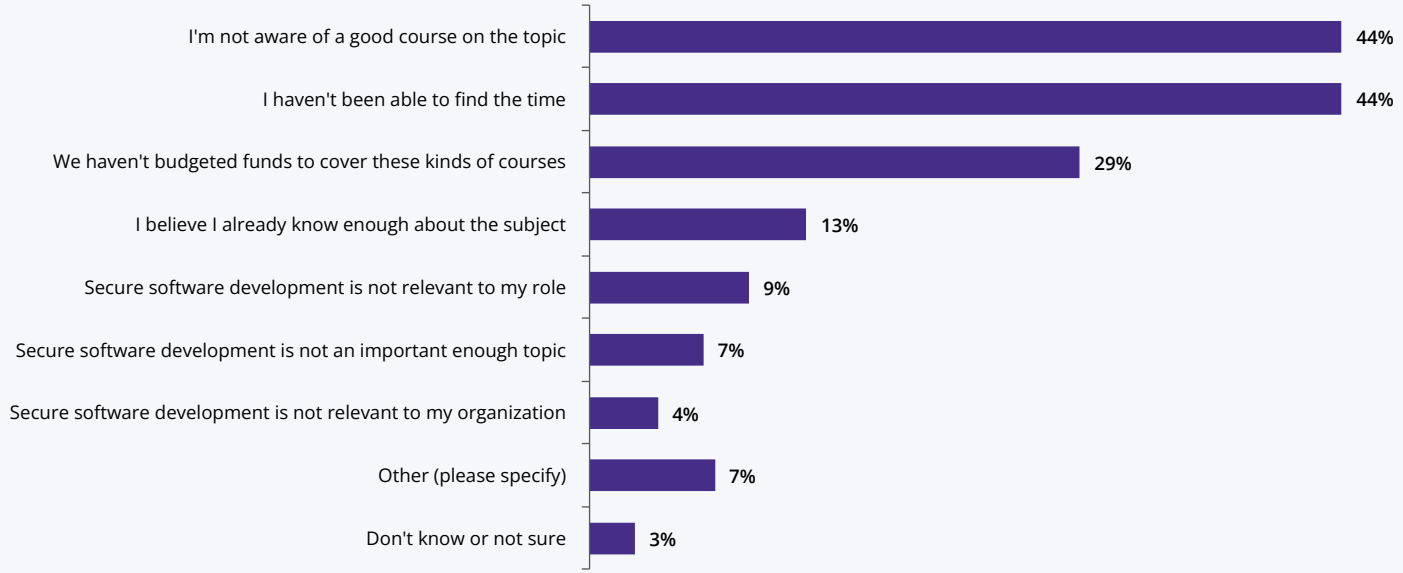
Respondents have not taken a course because they are not aware of a good one

The top reason for not taking a course in secure software development is being unaware of a good one, as depicted in Figure 7. This finding has several implications. First, budget is not the primary constraint, as only 29% of the respondents report this reason. Second, few respondents cited reasons implying that they don't want or need such training, such as believing that they know enough about the subject (13%), the subject not being relevant to their role (9%), the subject not being important enough (7%), or the subject not being relevant to their organization (4%).

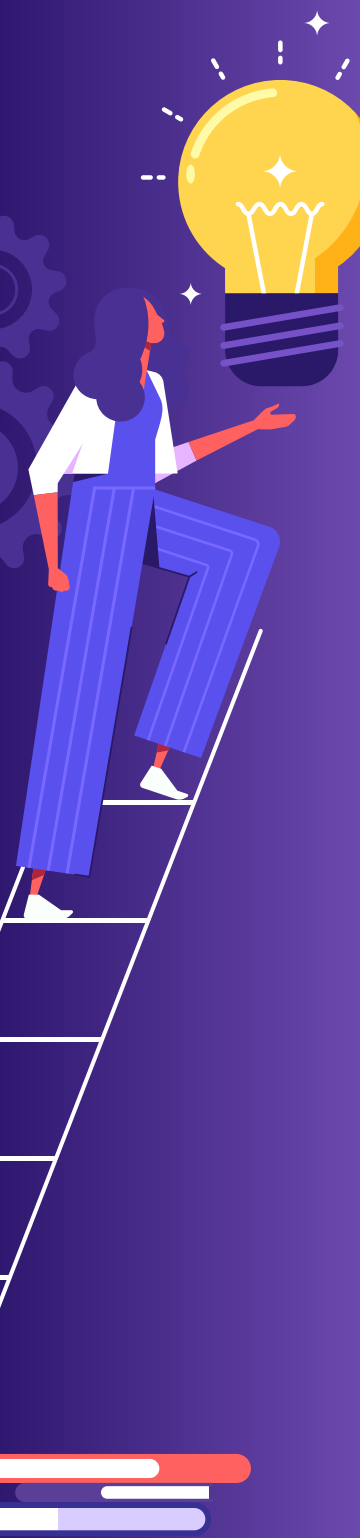
In Figure 7, we also notice that time constraints appear among the top challenges very close to not being aware of a good course on the topic. This finding reflects the tight schedules that software development professionals face. Any training in the area should be flexible and objective, allowing professionals to learn without disrupting their productivity.

FIGURE 7

REASONS FOR NOT TAKING A COURSE IN SECURE SOFTWARE DEVELOPMENT



2024 SecEd Survey, Q31, Sample Size = 150, Total Mentions = 239, question answered only by those who answered "No" in Q20



Respondents are unaware that OpenSSF offers free educational material

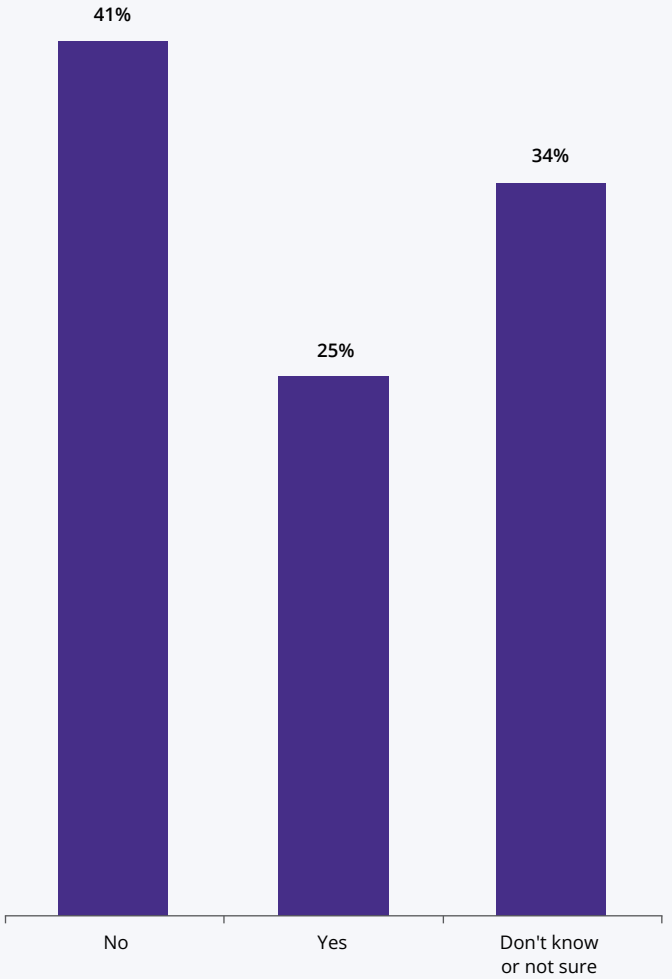
Many organizations offer training specifically for secure software development, including the OpenSSF. OpenSSF is a collaborative initiative hosted by the Linux Foundation to improve the security of OSS. Among many initiatives, OpenSSF offers training programs, educational materials, and resources to equip developers with the knowledge and skills necessary for secure coding. OpenSSF even offers a free course on the fundamentals of developing secure software. However, as pointed out in Figure 8, only one-quarter of the respondents report that their organizations use these materials.

OpenSSF is a collaborative initiative hosted by the Linux Foundation to improve the security of OSS. Among many initiatives, OpenSSF offers training programs, educational materials, and resources to equip developers with the knowledge and skills necessary for secure coding.

The main reason for not using any material is not being aware that OpenSSF offers such materials, as shown in Figure 9. OpenSSF, aware of this research's results, decided to provide more training on secure software development and intensify its advertising efforts.

FIGURE 8

PERCENTAGE OF ORGANIZATIONS THAT USE SECURE SOFTWARE DEVELOPMENT EDUCATIONAL MATERIALS FROM OPENSSF



2024 SecEd Survey, Q21, Sample Size = 398

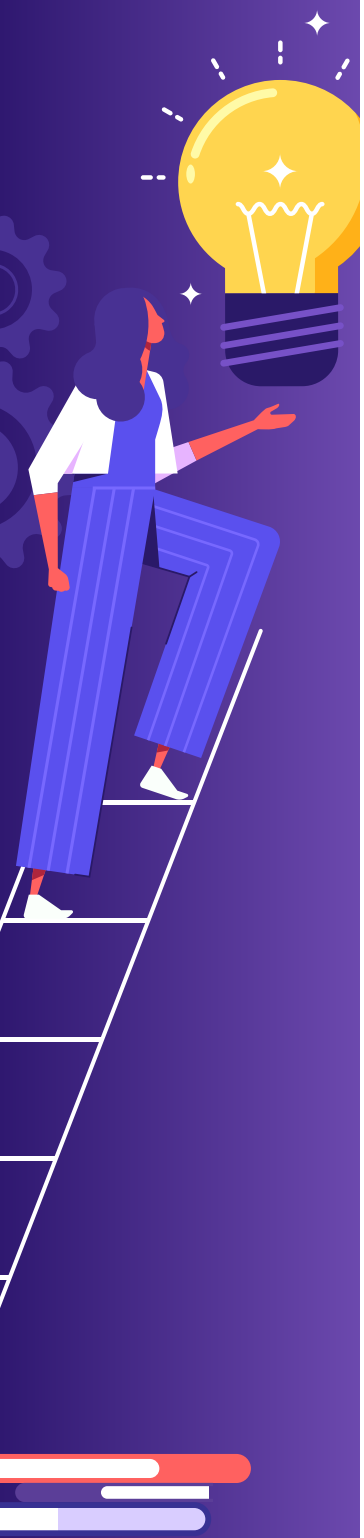
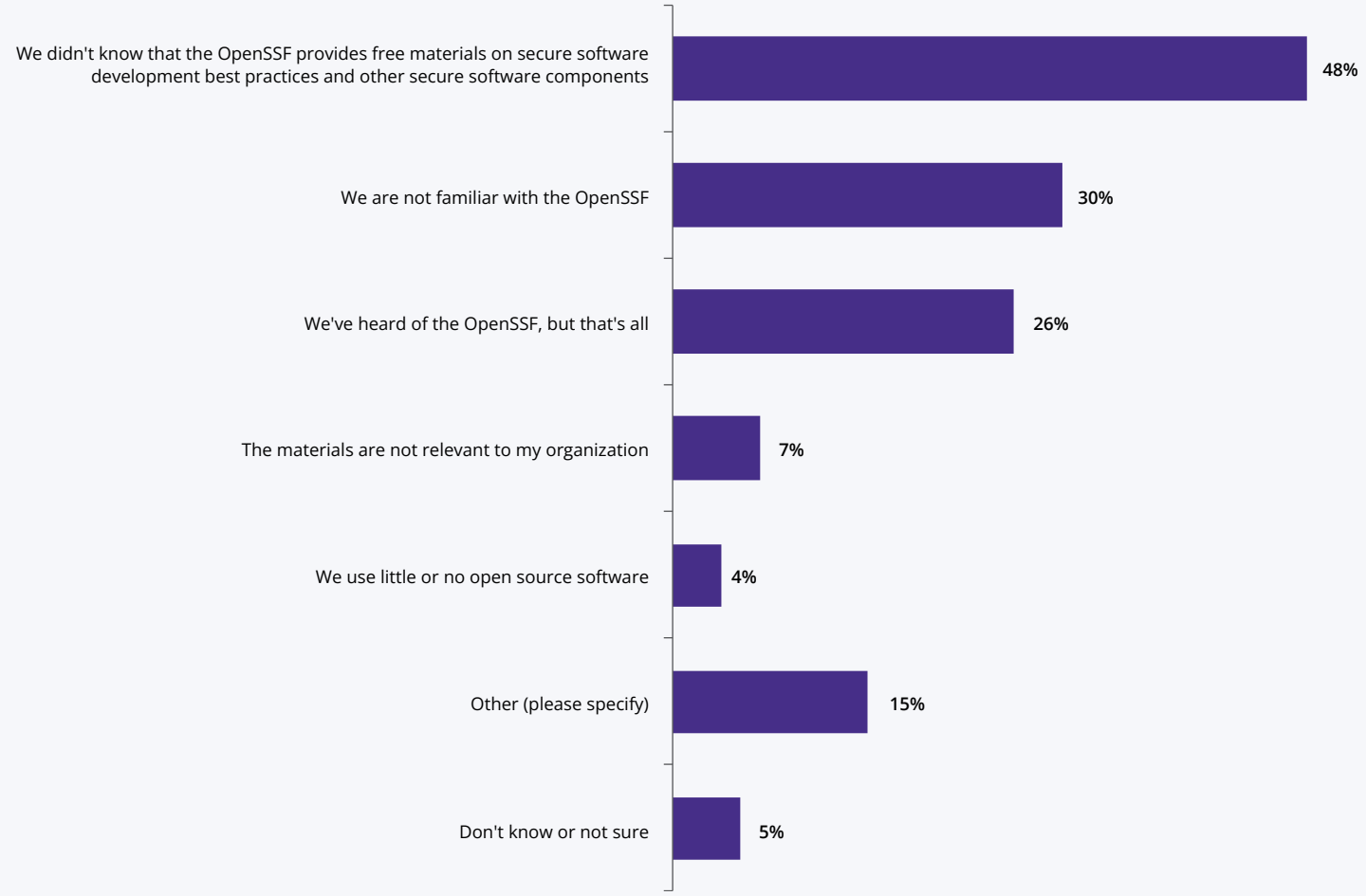
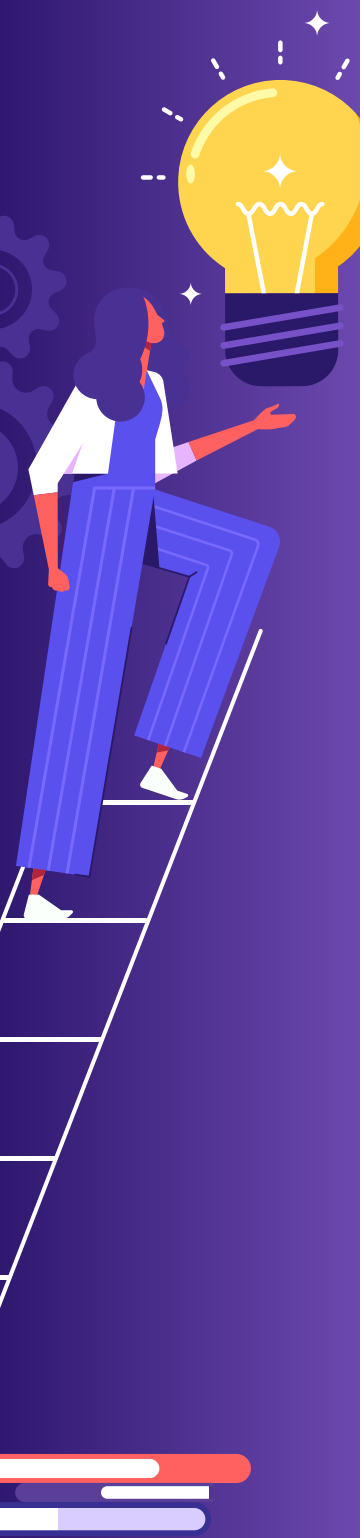


FIGURE 9

REASONS FOR NOT USING EDUCATIONAL MATERIALS FROM OPENSSF



2024 SecEd Survey, Q32, Sample Size = 135, Total Mentions = 181, question answered only by those who answered “No” in Q32

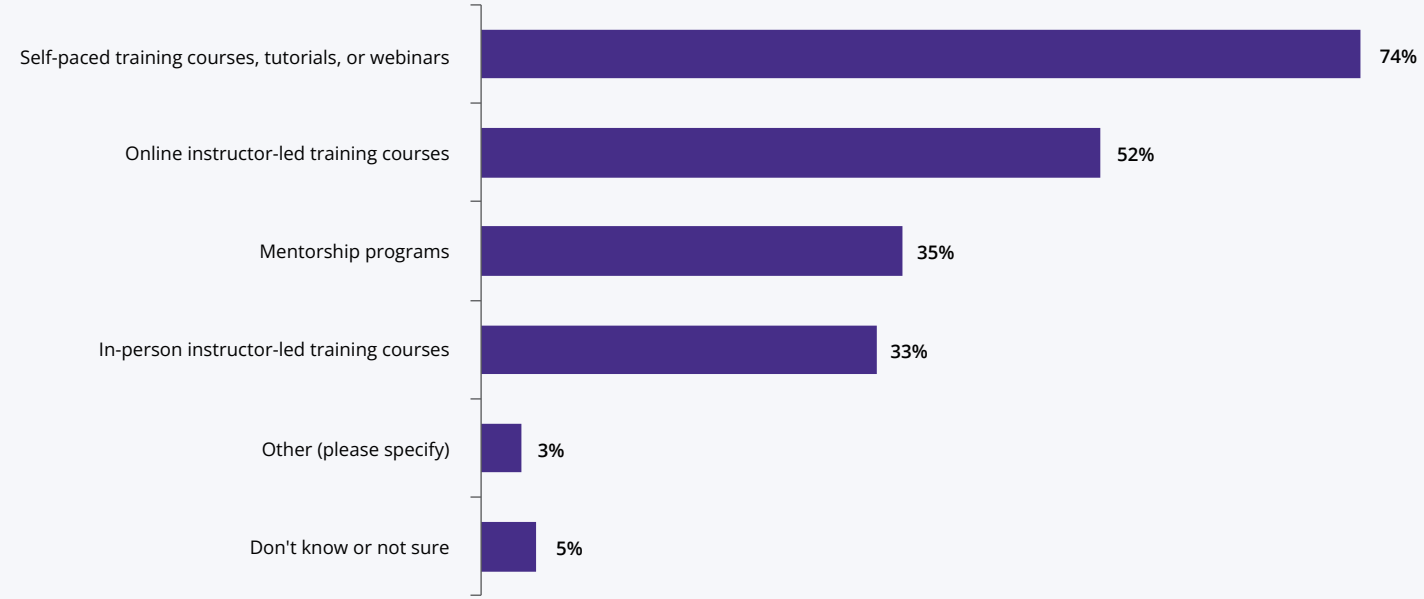


Respondents prefer self-paced training

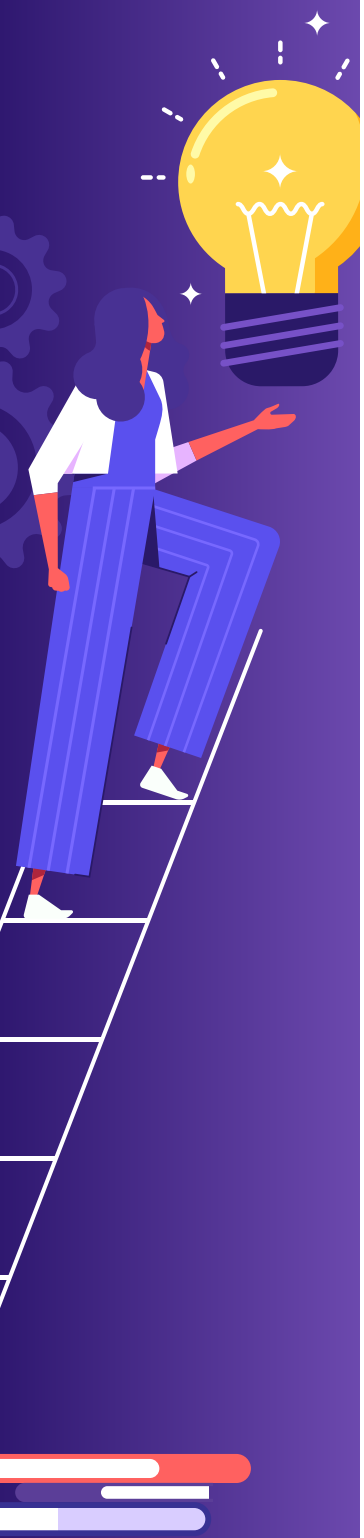
The preferred training option by the organizations is a self-paced approach, with 74% of respondents indicating its usefulness, as noted in Figure 10. This preference reflects the need for flexible learning opportunities that fit busy schedules. Online instructor-led training courses are also highly valued, with 52% of respondents finding them useful, suggesting a demand for more interactive and structured learning experiences. Mentorship programs are preferred by 35% of respondents, indicating the importance of personalized guidance and support in mastering security skills. Additionally, 33% of respondents see in-person instructor-led training courses as beneficial, emphasizing the value of face-to-face learning environments. These insights underline the diverse preferences for security education formats and the necessity for organizations to offer a variety of training options to meet different learning needs.

FIGURE 10

MOST USEFUL SECURITY-FOCUSED EDUCATION PROGRAMS OR RESOURCES



2024 SecEd Survey, Q30, Sample Size = 324, Total Mentions = 658



Chapter 3: Priority areas for training

As described in the previous chapter, there is a generalized need for more training in secure software development. Given the broad scope of this field, it is essential to understand how to prioritize training efforts and the development of new courses. This chapter explores this perspective by analyzing whether participants prefer language-agnostic or language-specific training and identifying the most needed topics for a course.

The main findings of this chapter are as follows:

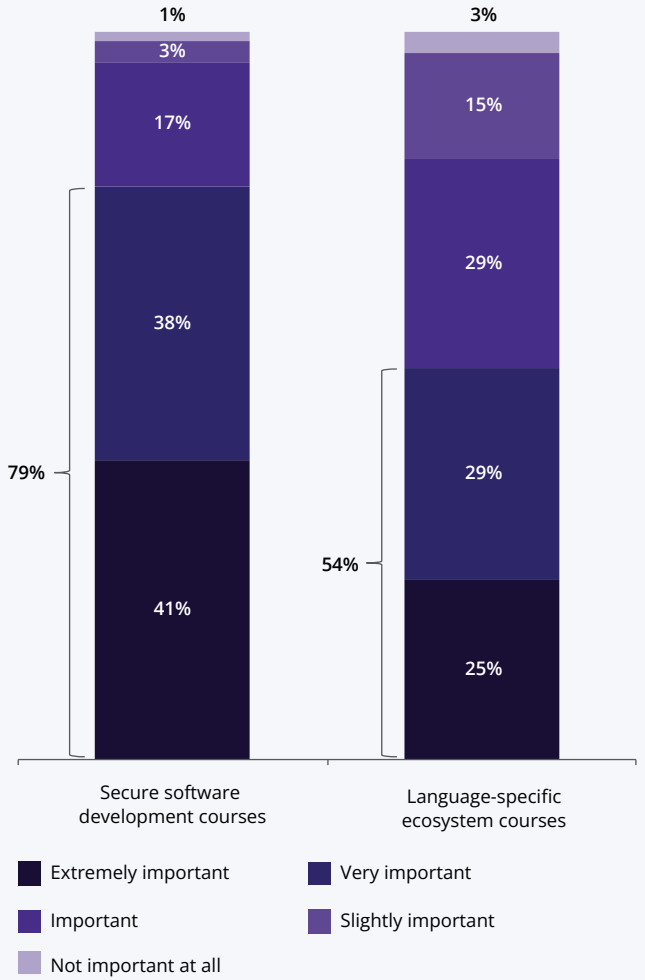
1. 79% of respondents consider language-agnostic courses highly important, compared with 54% who attribute a similar level of importance to language-specific courses.
2. The higher level of importance attributed to language-agnostic courses is consistent across various roles, involvement with OSS, regions, types of companies, and organization sizes.
3. Organizations require a diverse range of language-agnostic courses to enhance their IT staff's capabilities in secure software development, with security architecture (64%) emerging as the most popular choice among respondents, closely followed by security education and guidance (64%) and secure implementation (63%).
4. There is a large variation in training needs according to the professional role, OSS involvement, and years of experience, and the most popular choice can be security architecture (software developers and system operations), secure implementation (management and data science), threat assessment (security team), or policy and compliance (OSPO team).
5. Overall, respondents ranked security education and guidance as their top priority (but see below for caveats on this ranking).
6. A Python-specific course is in high demand among respondents, with 71% favoring it, while JavaScript (client side), the second place, is favored by 49% when relative ranking among languages was not considered.
7. Python emerges as the most requested course across all subpopulations, except for OSS committers. This group reports a higher need for C courses, though Python remains a close second in their preferences.
8. Despite Python's overall popularity, when participants were asked to rank their choices, C (22%) and Java (18%) were selected as the top choice more frequently than Python (17%).
9. Respondents also report a variety of courses needed by their organizations, emphasizing the importance of specialized training in certifications, testing, secure coding practices, and supply chain security.
10. Looking forward, AI and ML security is the primary area needing increased attention and innovation, identified by 57% of respondents, with supply chain security closely following, selected by 56% of participants.

Professionals consider language-agnostic training more important than training focused on a specific language

As depicted in Figure 11, 79% of respondents consider programming language–agnostic secure software development training extremely or very important, compared with 54% who view programming language–specific training with this level of importance. Programming language–agnostic courses on secure software development offer several advantages over their language-specific counterparts. Firstly, they provide a broad understanding of security principles that apply across various programming languages and platforms, enabling learners to apply these concepts across different ecosystems. Language-agnostic courses emphasize foundational security practices such as threat modeling, secure design principles, and risk assessment, which are critical skills irrespective of the specific programming language used. This universality not only makes the knowledge more versatile and applicable in diverse work settings but also prepares developers for future technologies and languages that may emerge.

FIGURE 11

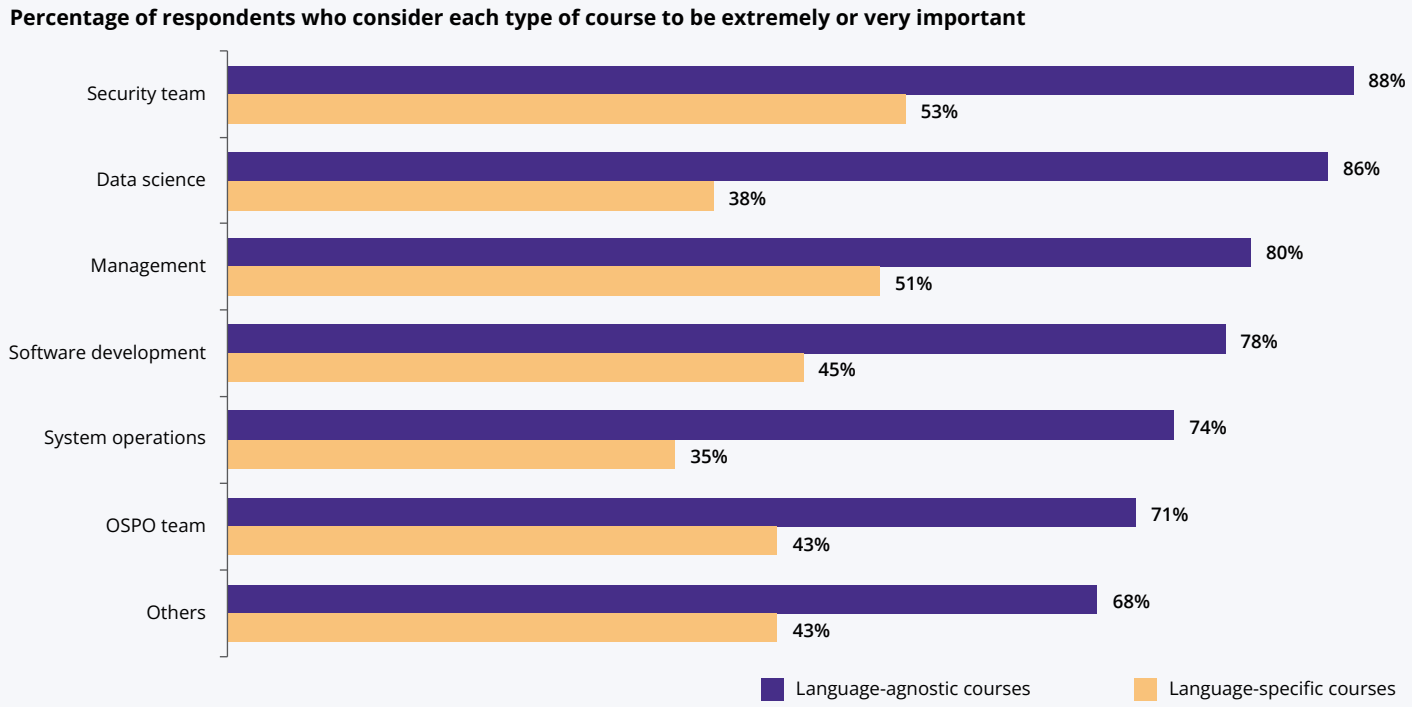
IMPORTANCE OF LANGUAGE-AGNOSTIC SECURE SOFTWARE DEVELOPMENT COURSES AND LANGUAGE-SPECIFIC ECOSYSTEM COURSES



2024 SecEd Survey, Q27, Sample Size = 316, DNKS excluded

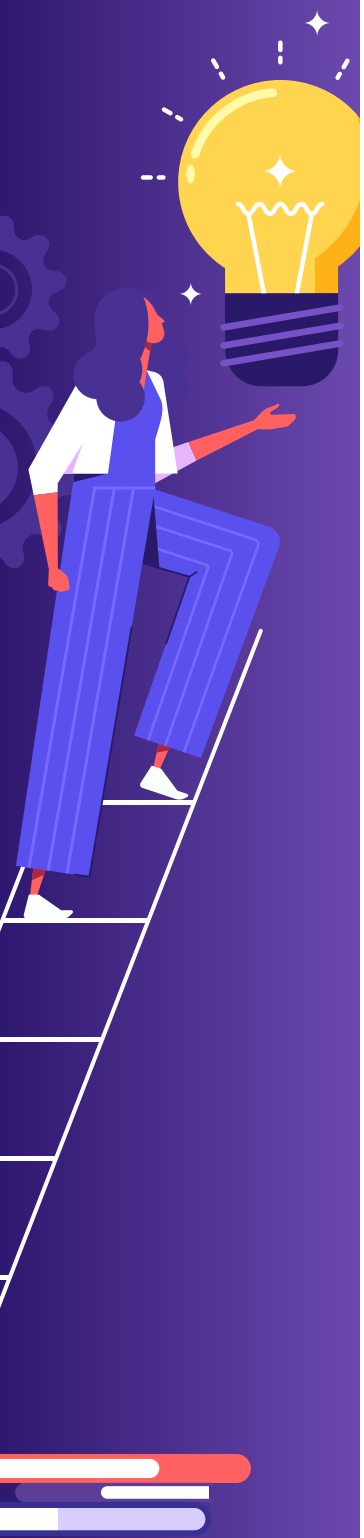
FIGURE 12

COMPARISON OF THE LEVEL OF IMPORTANCE OF EACH TYPE OF COURSE, SEGMENTED BY EACH ROLE



2024 SecEd Survey, Q27 by Q5, Sample Size = 316 for language-agnostic and 318 for language-specific, DNKS excluded

Figure 12 indicates that the preference for language-agnostic courses is consistent across roles, with the strongest preference coming from security team members, who often need to address systems developed in multiple languages. Additionally, we analyzed differences segmented by contributions to OSS, OSS roles, regions, types of companies, and organization sizes. In all these segments, respondents consistently rated language-agnostic courses as more important than language-specific ones.



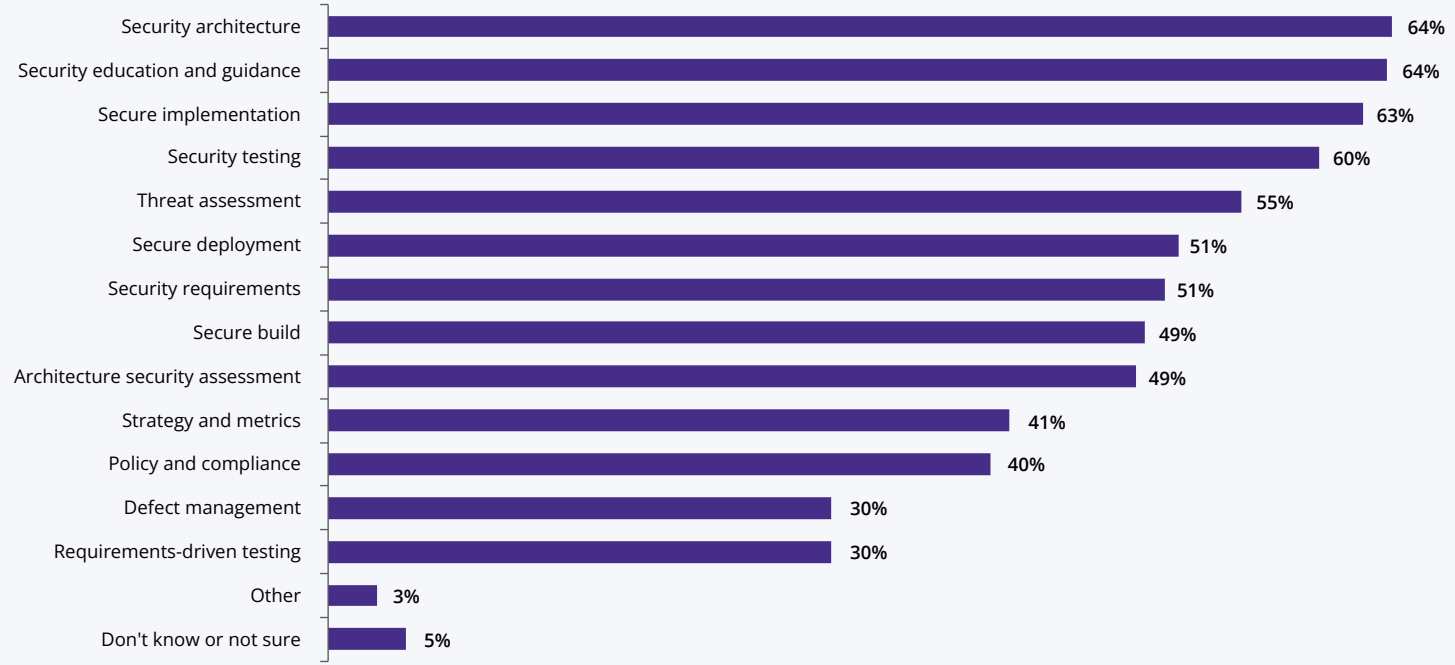
Organizations need a great variety of language-agnostic courses, and security architecture is the most popular

Organizations need a variety of language-agnostic courses to fill educational gaps and help IT staff better address secure software development. As observed in Figure 13, nine courses were selected by at least 49% of the respondents: secure architecture, security education and guidance, secure implementation, security testing, threat assessment, secure deployment, security requirements, secure build, and architecture security assessment. The most popular choice for our respondents was security architecture (64.3%), closely followed by security education and guidance (64.0%) and secure implementation (62.6%).

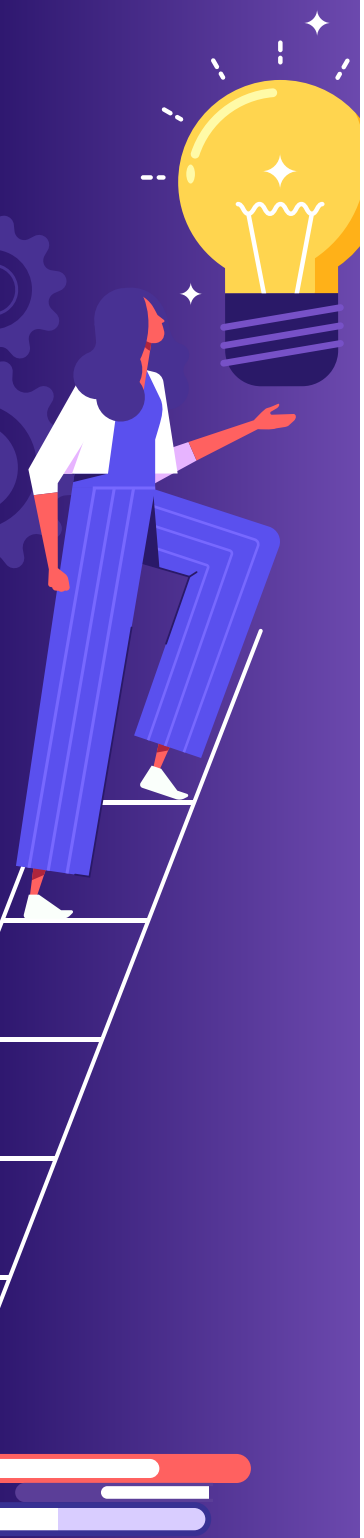
Security architecture provides a structured framework that defines the processes, tools, and protocols required to create and maintain secure software systems. A security architecture

FIGURE 13

LANGUAGE-AGNOSTIC COURSES COULD FILL SIGNIFICANT KNOWLEDGE GAPS FOR ORGANIZATIONS' IT STAFF TO BETTER ADDRESS SECURE SOFTWARE DEVELOPMENT



2024 SecEd Survey, Q25, Sample Size = 342, Total Mentions = 2,244



enables the consistent application of security standards across all projects, serving as a blueprint for implementing security measures that align with organizational goals and compliance requirements. Such a course would cover how to address security concerns associated with components and technology during the architectural design, development, and deployment stages of software to meet security requirements.

The purpose of security education and guidance is to "provide training for employees to increase their security awareness and leverage this knowledge and other guidance in the design, development, and deployment of secure software." In retrospect, this option should have been more clearly defined, as this had more than one interpretation. One interpretation is that its purpose was to help organizations determine how to devise training sequences for employees that would be most relevant. We believe many respondents did not interpret the question in this way, as increasing experience lowered the likelihood of this choice (the opposite of what one might expect). An alternative interpretation would be that this was asking for "fundamentals" focusing on general knowledge about security education and guidance. We believe, given the other data, that this was the interpretation most respondents intended. It's worth noting that the OpenSSF already has a course on the fundamentals of developing secure software, but as also noted earlier, many respondents were unaware of it.

Finally, secure implementation in software development involves writing source code to avoid common vulnerabilities and be more robust against attacks. This approach ensures another level of defense, ensuring that security is embedded in the code of software products from the outset. During secure implementation, developers apply secure coding practices to prevent vulnerabilities such as SQL injection, cross-site scripting, and buffer overflows. The objective is to mitigate risks early in the development cycle, reducing the cost and complexity of fixing security issues after deployment. Note that while the fundamentals of secure implementation can be taught without being specific to a programming language, more advanced topics generally do require focusing on specific languages.

Different roles have different needs

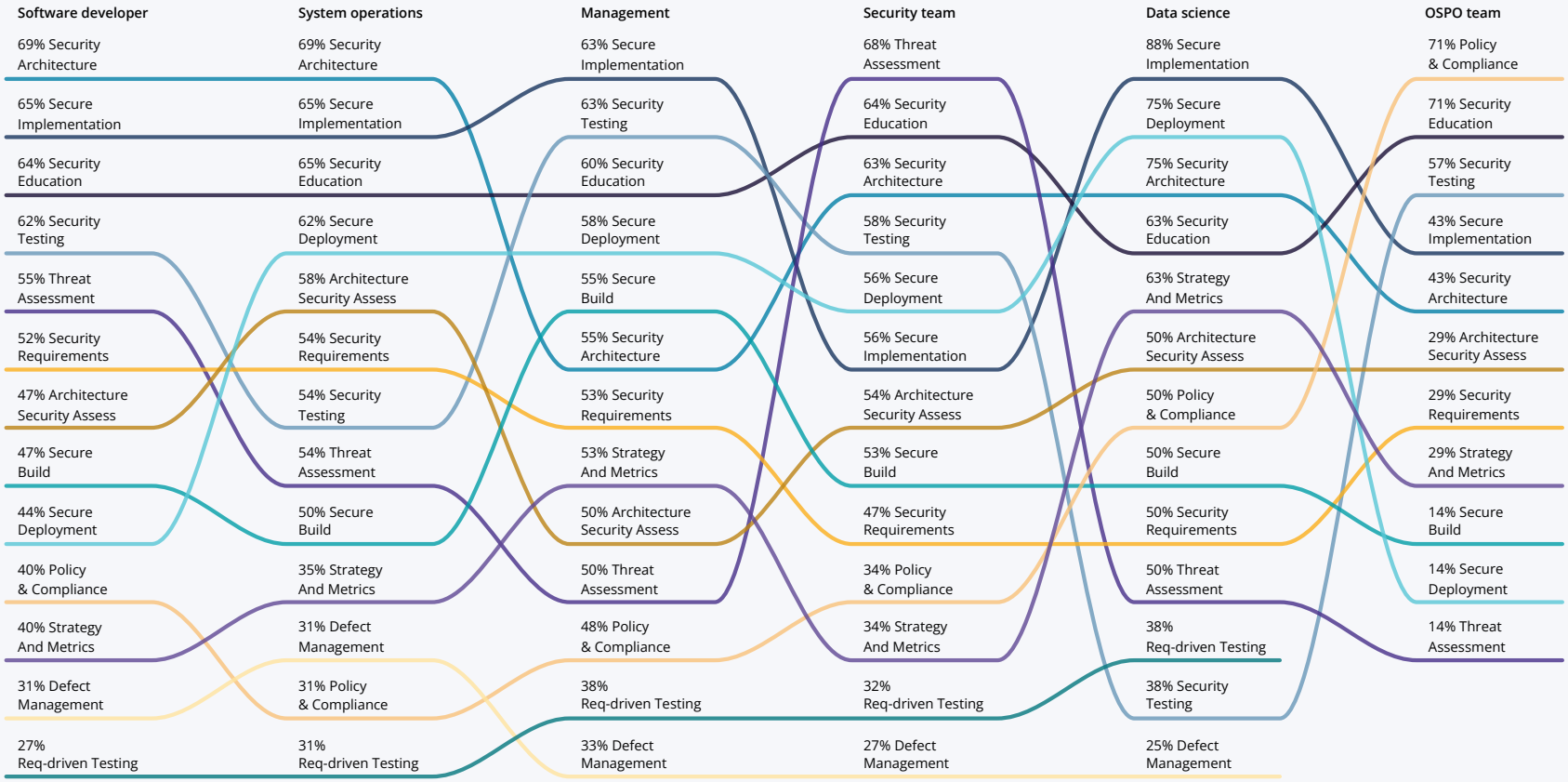
As Figure 14 shows, there is considerable variation in the training needs reported by each role. Security architecture emerges as the most popular course among software developers and operations personnel, who are directly involved in software development and deployment. However, it ranks lower for managers (sixth), security teams (third), data science professionals (third), and OSPO teams (fifth). Secure implementation is the preferred course for managers and is especially popular with data science professionals. For security teams, threat assessment ranks as the most relevant course, while for OSPO team members, policy and compliance is the top priority.

We also segmented this analysis from multiple perspectives, detailed in Appendix B with complete rankings. For levels of contribution to OSS (Figure 27), region (Figure 29), type of organization (Figure 30), organization size (Figure 31), and familiarity (Figure 32), there is some variation in the top positions, usually held by secure architecture or security education and guidance. However, the percentage of respondents does not vary considerably in these cases. Conversely, the OSS role (Figure 28) appears to influence preferences for educational courses, similar to professional roles (Figure 14). Additionally, years of experience also seem to affect course selection, with security testing highly ranked among those with less than five years of experience, security architecture for those with five to 20 years, and secure implementation for those with over 20 years.

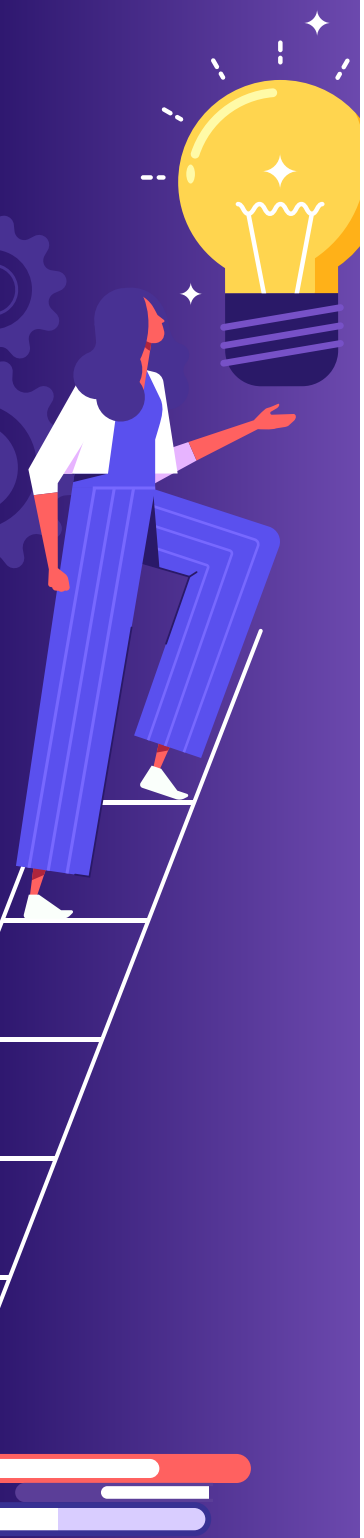
FIGURE 14

RANKINGS OF POPULARITY FOR THE LANGUAGE-AGNOSTIC COURSES, SEGMENTED BY ROLE

Which of the following courses could fill significant gaps for the organization you work for to help IT staff better address secure software development? (select all that apply)



2024 SecEd Survey, Q25 by Q5, Sample Size = 312, Total Mentions = 2,035, the number in front of the name represents the percentage of respondents, each column is sorted by this number

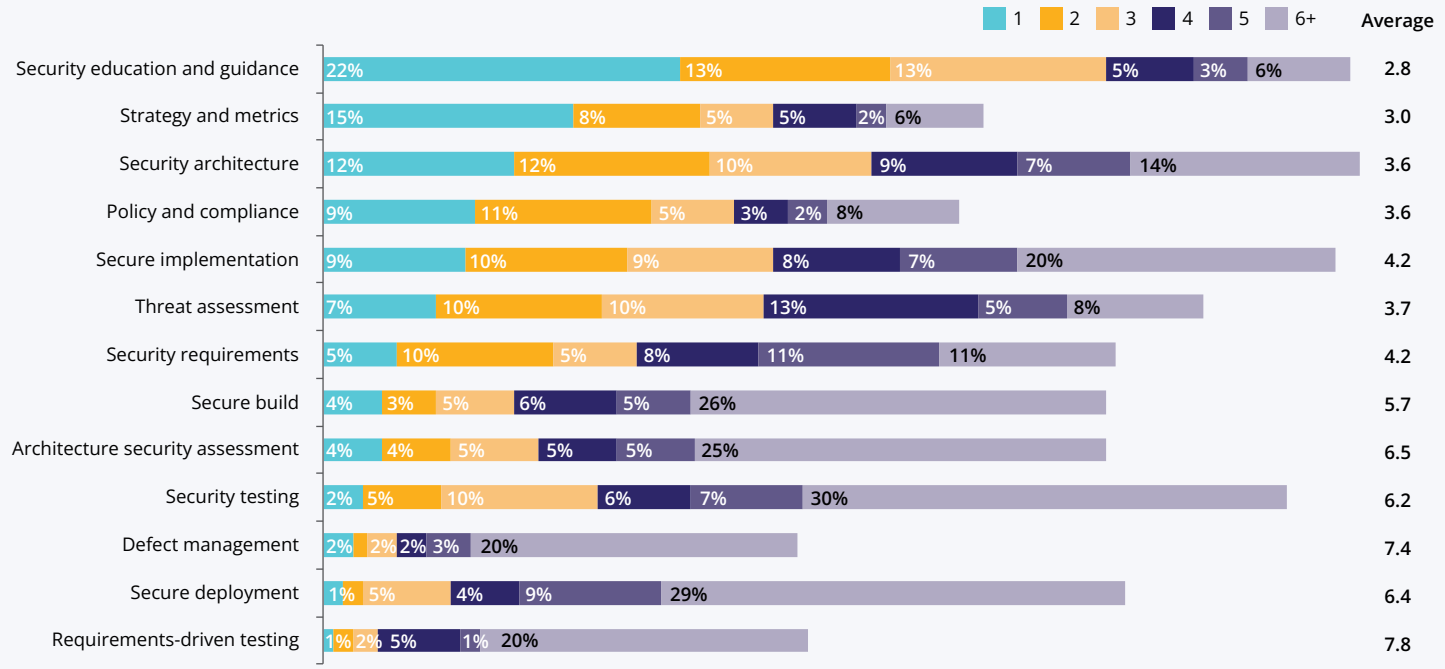


Respondents consider security education and guidance their top priority

We also asked respondents to rank their choices for the courses shown in Figure 15 based on their importance. Figure 15 reveals that security education and guidance is most frequently chosen as the most important course. It maintains its lead in the rankings across the top five choices. Additionally, even when considering the average rankings, displayed on the right-hand side of Figure 15, this course continues to outrank others. Notably, sorting by average ranking (shown on the right-hand side of Figure 15) does not considerably alter the order of the figure, which is sorted by the percentage of first-place rankings.

FIGURE 15

IMPORTANCE ORDER ATTRIBUTED BY THE RESPONDENTS FOR EACH OF THE SELECTED LANGUAGE-AGNOSTIC COURSES



2024 SecEd Survey, Q26, Sample Size = 308, sorted by the percentage of first-place rankings

Figure 15 evidences strategy and metrics in the second position for both top choices and average rankings, despite its lower 10th place in terms of popularity, as shown in Figure 13. This disparity suggests that while strategy and metrics may not attract widespread attention, those who prioritize it find considerable value in its content, implying that the course is highly valued by those with specific needs that require deeper strategic and metrics-driven insights. Such insights could help educational providers to tailor and market this course more effectively to its most appreciative audience. Our data suggests that those who rank strategy and metrics as a top choice are predominantly from larger organizations (20,000+ employees) and possess a high degree of familiarity with secure software development. This trend suggests that individuals in more complex organizational environments, who have moved beyond basic security concepts, tools, and processes, require specific strategies and metrics to effectively develop and assess secure software.

A Python-specific course is a popular demand

Among the language-specific courses, there is a demand for Python-focused education, with 71% of respondents indicating this preference when ignoring rankings, as shown in Figure 16. This demand significantly exceeds that for the next most popular course, JavaScript (client side), which 49% of respondents favor. Even when combining the figures for those who selected JavaScript for both client side and server side, the JavaScript total only marginally increases to 53%, still considerably lower than Python. This is primarily due to a large overlap among those who use JavaScript on the client and those who use it on the server.

Python is the second most popular language on GitHub, trailing only behind JavaScript, and its use has surged by over 22% year over year. The language is also prominent in rapidly growing fields such as AI and ML.

Python's popularity could stem from several factors. It is known for being accessible to beginners. It has become a go-to language for many professionals, whether they are shifting from other languages or have been introduced to it during their educational journey. Notably, Python is the second most popular language on GitHub[5], trailing only behind JavaScript, and its use has surged by over 22% year over year. The language is also prominent in rapidly growing fields such as AI and ML. The shift of developers toward Python, combined with a relative scarcity of educational materials focused on secure software development for Python, likely contributes to the high demand for such courses. Some common vulnerabilities that plague Python code are injection and arbitrary command execution, insecure file handling, outdated dependencies, directory traversal, and improper package management.

Nevertheless, JavaScript, the second choice, remains an important part of software development, recognized by GitHub as the most common language in its repositories. Client-side JavaScript is exposed to vulnerabilities such as cross-site scripting and sensitive data exposure.

5 https://github.blog/2023-03-02-why-python-keeps-growing-explained/

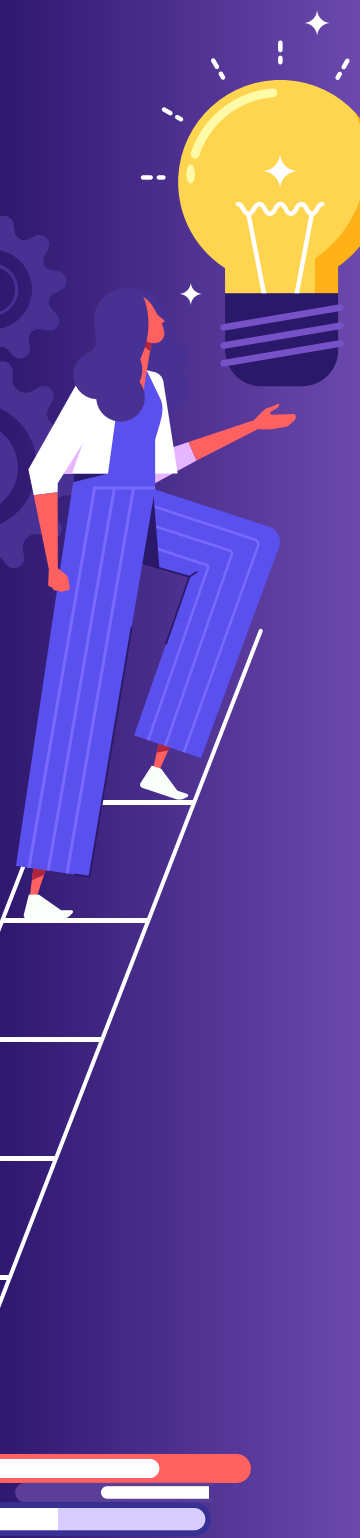
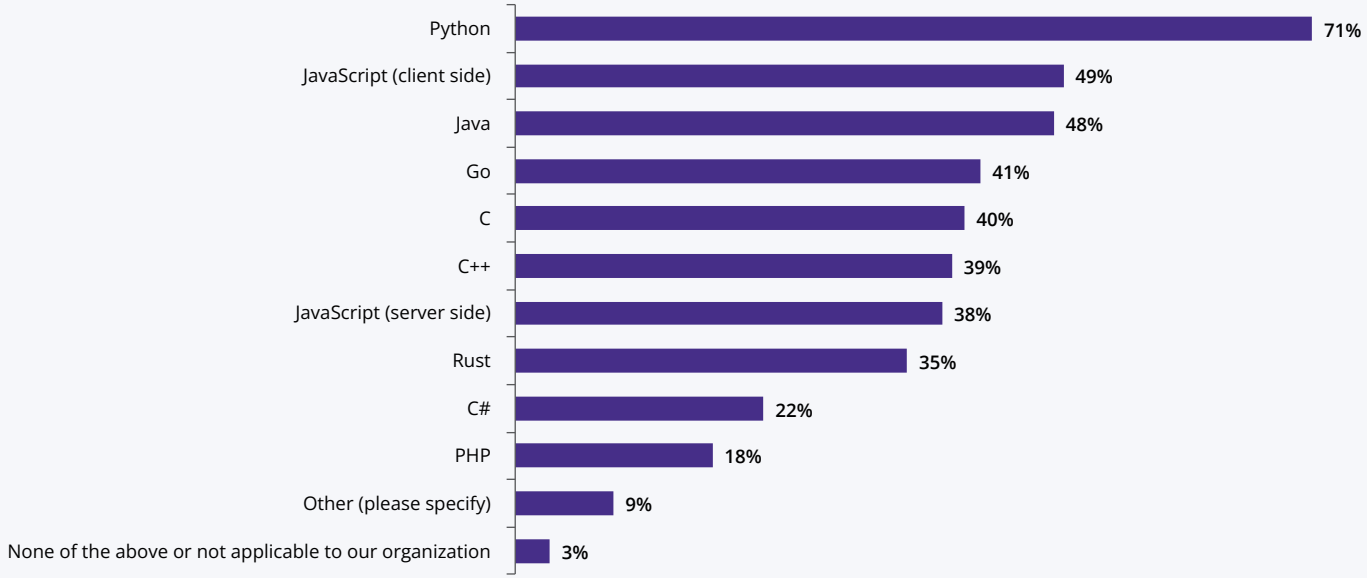


FIGURE 16

LANGUAGE-SPECIFIC ECOSYSTEM COURSES THAT ORGANIZATIONS SHOULD MAKE AVAILABLE TO THEIR DEVELOPERS



2024 SecEd Survey, Q23, Sample Size = 352, Total Mentions = 1,454

The popularity of Python is confirmed across different populations

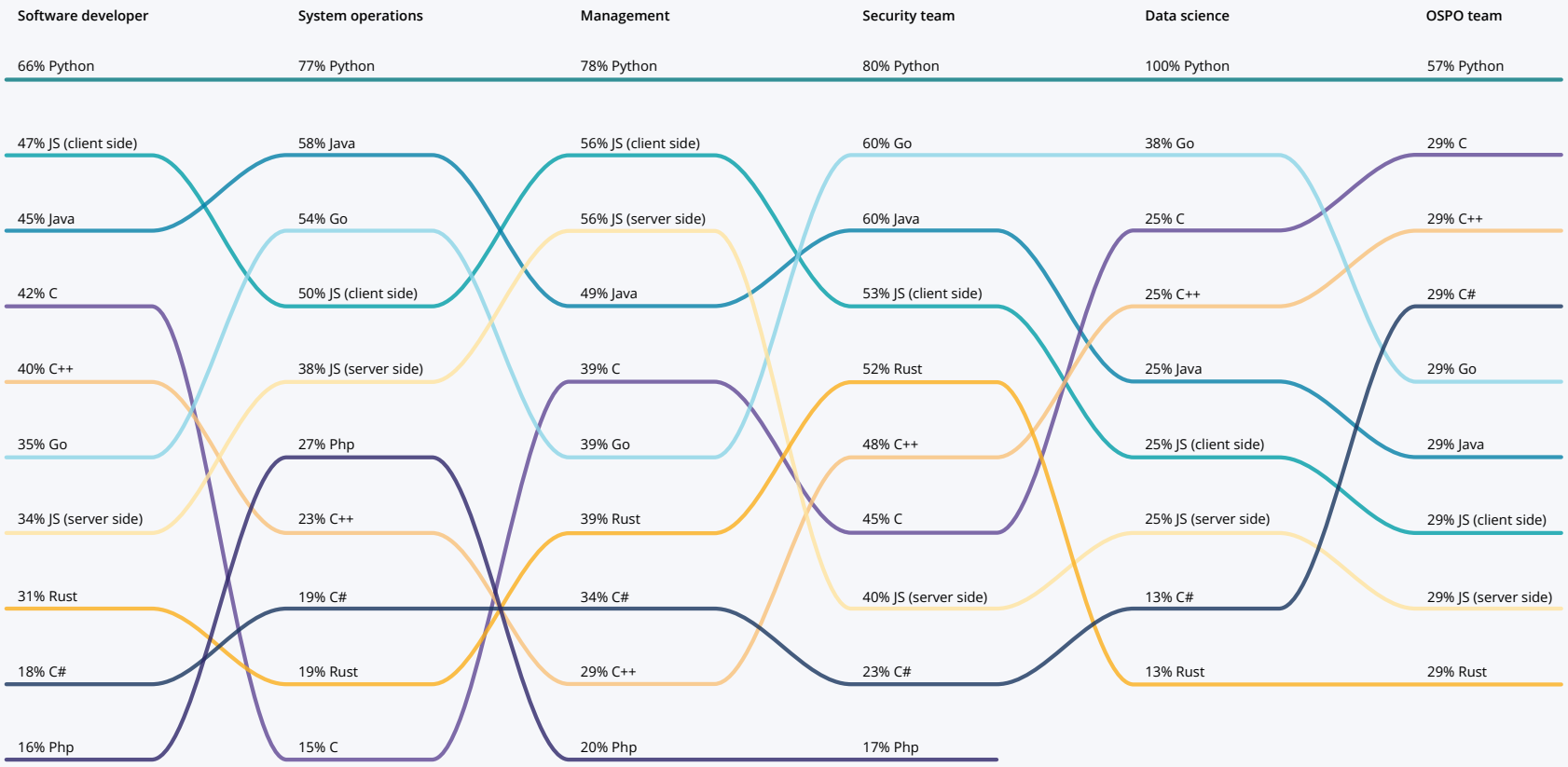
As illustrated in Figure 17, a Python-specific security course emerges as the most popular language-specific demand across all respondent roles when ignoring rankings. Figure 17 also confirms Python's popularity in data science, with participants in these roles universally recognizing the relevance of Python-specific courses to their needs.

As observed in Appendix C, which segregates rankings for language-specific courses, Python emerges as the most requested course across all subpopulations except for OSS committers. This group reports a higher need for C courses, though Python remains a close second in their preferences. This highlights the specific demands of OSS committers, who may deal more frequently with lower-level programming challenges. In contrast, the broader popularity of Python highlights its widespread utility and appeal in various fields.

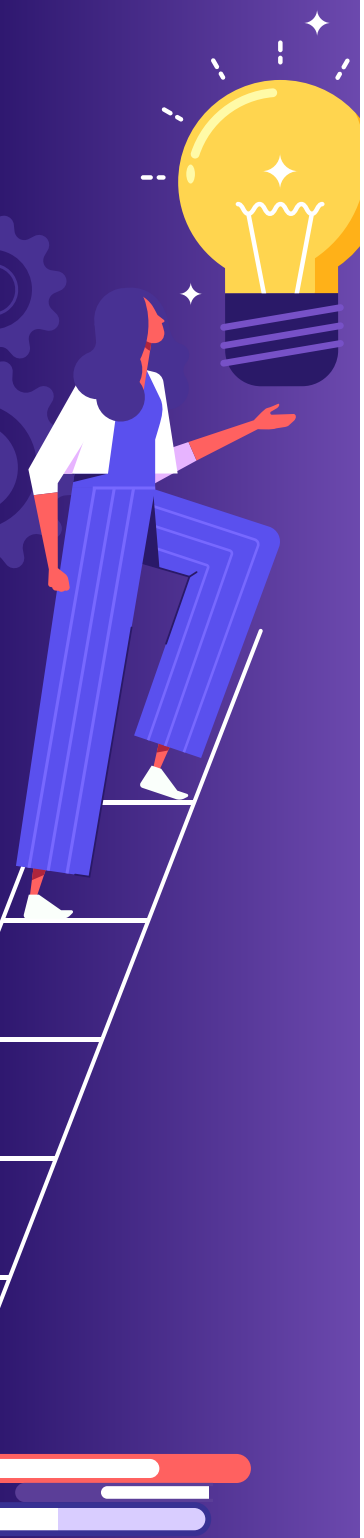
FIGURE 17

RANKINGS OF POPULARITY FOR THE LANGUAGE-SPECIFIC COURSES, SEGMENTED BY ROLE

Which language-specific ecosystem course(s) on secure software development should the organization you work for make available to its developers? (select all that apply)



2024 SecEd Survey, Q23 by Q5, Sample Size = 321, Total Mentions = 1,320, the number in front of the name represents the percentage of respondents, each column is sorted by this number



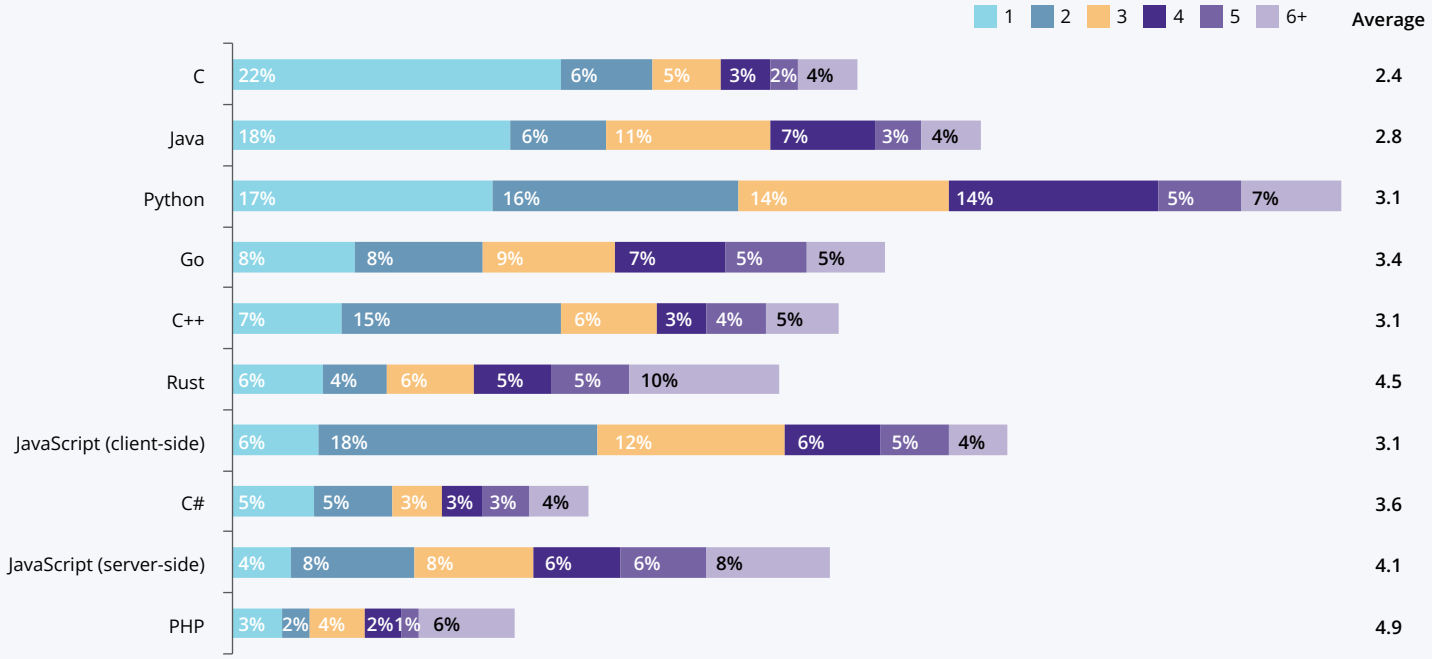
C and Java are more frequently selected as top-choice courses

In contrast to its overall popularity, when participants were asked to rank their choices, C and Java were often selected as the top choices more frequently than Python, as shown in Figure 18.

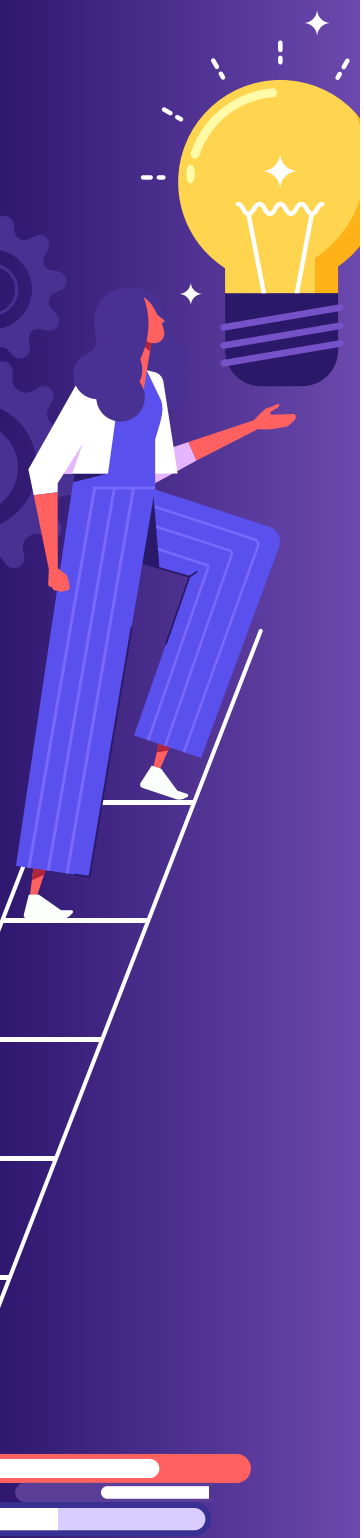
However, Python leads when considering the top two or three choices. A possible explanation is that Python is often chosen together with other languages, indicating its role in a bigger ecosystem of programming languages, whereas C and Java are more commonly selected on their own.

FIGURE 18

IMPORTANCE ORDER ATTRIBUTED BY THE RESPONDENTS FOR EACH OF THE SELECTED LANGUAGE-SPECIFIC COURSES



2024 SecEd Survey, Q24, Sample Size = 331, sorted by the percentage of first-place rankings



C is used to build critical infrastructure, and there is certainly much to explore in secure software development courses focused on this language. C is susceptible to vulnerabilities such as buffer overflows, uninitialized variables, null pointer dereferencing, improper type conversions, use after free, and double frees. Java, another versatile programming language, is widely used across various types of systems and applications. Despite a slight decrease in popularity, as measured by TIOBE[6], it remains pivotal in creating a variety of new systems, and there are many production systems in the market using Java, given the decades of the language's existence.

Respondents report a variety of courses needed by their organization

Respondents could also freely report courses they need in open-ended questions. Table 1 presents the classification of their answers. While many of the options are already explicitly covered in the options above, some interesting recurrent topics emerged.

The table highlights the value of certifications, which authenticate the expertise of professionals in specific topics and establish a standardized knowledge base among professionals. For organizations, certifications ensure that individuals handling software security are well versed in best practices and the latest methodologies. Certifications such as Certified Secure Software Lifecycle Professional (CSSLP) and Certified Ethical Hacker (CEH) validate an individual's expertise and commitment to the field and are often required by employers. Moreover, certifications help maintain a continual learning culture, as they often require ongoing education and renewals, encouraging professionals to keep abreast of evolving threats and technologies.

Verification was also a recurring theme among the responses. Regular and comprehensive verification not only helps avoid bugs that can potentially be exploited by attackers but also can exercise specific security aspects. Static application security testing (SAST) examines source code to find vulnerabilities without executing it. Unit testing provides specific inputs to parts of programs ("units") and then determines if the result is the expected one or not. Fuzzing "randomly" generates inputs and then executes software to detect undesired behavior (such as crashing). Web application scanners simulate an attacker's browser, crawling through a web application's web pages and examining it for security vulnerabilities. The term "dynamic application security testing" (DAST) has various meanings: It is sometimes used as a synonym for web application scanners[7,8] while others use it to include web application scanners and fuzzers.[9,10] Together, such techniques catch a broad range of vulnerabilities at various stages of the software development lifecycle. Importantly, some respondents reported the need for courses emphasizing security testing automation and its integration into the continuous integration/continuous deployment (CI/CD) pipeline.

6 https://www.tiobe.com/tiobe-index/

7 https://www.veracode.com/security/dast-test

8 https://www.csoonline.com/article/3487708/9-top-fuzzing-tools-finding-the-weirdest-application-errors.html

9 https://insights.sei.cmu.edu/sei_blog/2018/07/10-types-of-application-security-testing-tools-when-and-how-to-use-them.html

10 https://blog.code-intelligence.com/what-is-fast

The necessity of integrating security directly into the implementation process is emphasized, highlighting the importance of educational programs that focus on best implementation practices and defensive programming. Such courses teach developers to incorporate security measures right from the initial stages of software development, making code more robust against potential threats.

Courses related to supply chain security were also recurrently requested. Modern software does not exist in isolation but is connected to a vast network of interdependencies with external packages. The complexity and interconnectedness of modern software supply chains mean that vulnerabilities in any component can compromise the entire system, as evidenced by some recent major cybersecurity issues. Therefore, it's essential for developers to implement strict security practices throughout the supply chain. This includes vetting third-party vendors, using verified and secure open source libraries, and continuously monitoring for vulnerabilities in third-party components. Additionally, maintaining an accurate and up-to-date software bill of materials (SBOM) is crucial, as it provides transparency about all components used in the software, enabling better management of potential risks.

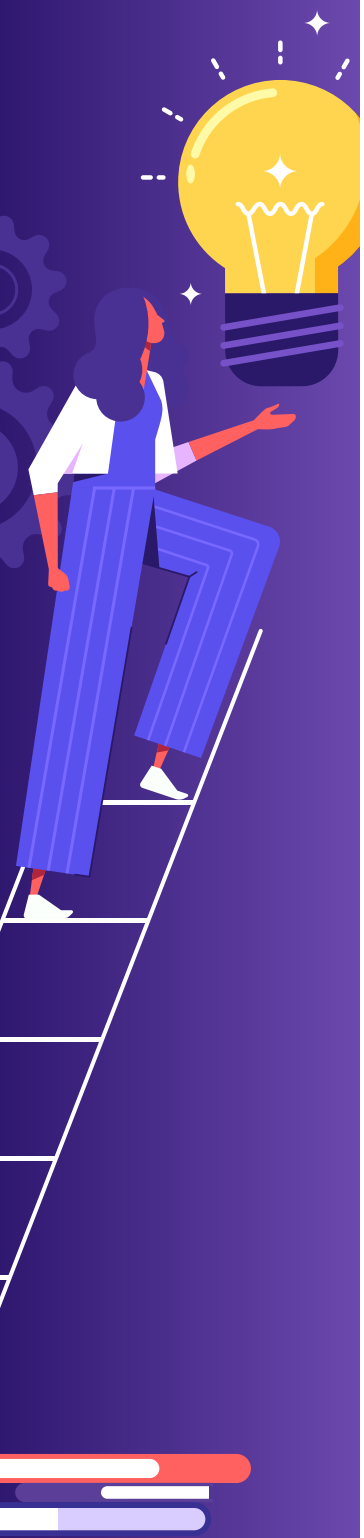
In sum, Table 1 categorizes recurrent secure software development education areas, emphasizing the need for specialized training tailored to the multifaceted challenges of IT security.

TABLE 1

RESPONDENT-RECOMMENDED COURSES IDENTIFIED IN AN OPEN-ENDED SURVEY QUESTION

Topic	Examples
Certification	CEH, CASE, CSSLP, CISA, CISSP, CSSE, CSSLP, OSP
Testing	Automated security testing, modern testing to legacy code, code security testing, DAST, SAST, fuzzing, penetration testing, unit testing
Coding practices	Best coding practices, coding rules, defensive programming, error handling, how to build security into code, coding based on OWASP top 10
Supply chain	Supply chain security, SBOM, dependency management, screening packages before use, Sigstore, supply chain attacks, tooling
Threat modeling	Agile threat modeling, threat analysis, threat intelligence, threat modeling with effective definition of trust boundaries, vulnerability analysis
Secure architecture	Secure by design, secure API development, secure design patterns, designing secure software with emphasis on testing
Cloud security	Cloud-native security best practices, cloud configurations, public clouds, AWS, AZ-500
Secure software development (general)	Secure software development for engineers early in their career, secure software development fundamentals, holistic security perspective
Identity and access management	Access control, access management, authentication, IAM

2024 SecEd Survey, Q22, Sample Size = 558, each participant provided two responses, table sorted by recurrence, only the top topics are shown



New areas may emerge in the future

Securing software development is a dynamic challenge due to its many components and the field's constant evolution. We surveyed our participants about the areas within secure software development that will require more attention and innovation in the future, as depicted in Figure 19.

AI and ML security has surfaced as a prominent concern, with 57% of survey respondents identifying it as an area needing heightened attention and innovation from the secure software development perspective. As these technologies become integral to various industries, their security implications grow more critical. The complexity of AI and ML systems, combined with their data-intensive operations, exposes them to unique vulnerabilities, such as data poisoning, model theft, and adversarial attacks. At this time, developing secure ML systems ("adversarial machine learning") involves many unsolved research problems, and currently known mitigations are typically weak against adversaries.[11] As these technologies continue to evolve and scale, robust AI and ML security practices will be increasingly critical.

Following closely behind AI and ML security, supply chain security was identified by 56% of respondents as a critical area, which is also in line with the results from the open-ended question discussed in the previous section. Securing supply chains will become increasingly important in the future due to the escalating complexity, interconnectivity, and globalization of software development ecosystems. As businesses integrate a multitude of third-party components and services—from libraries and frameworks to development tools—the attack surface for potential vulnerabilities expands significantly. Furthermore, as regulatory demands for software security and data protection grow stricter worldwide, compliance becomes more challenging.

Securing supply chains will become increasingly important in the future due to the escalating complexity, interconnectivity, and globalization of software development ecosystems.

11 https://csrc.nist.gov/pubs/ai/100/2/e2023/final

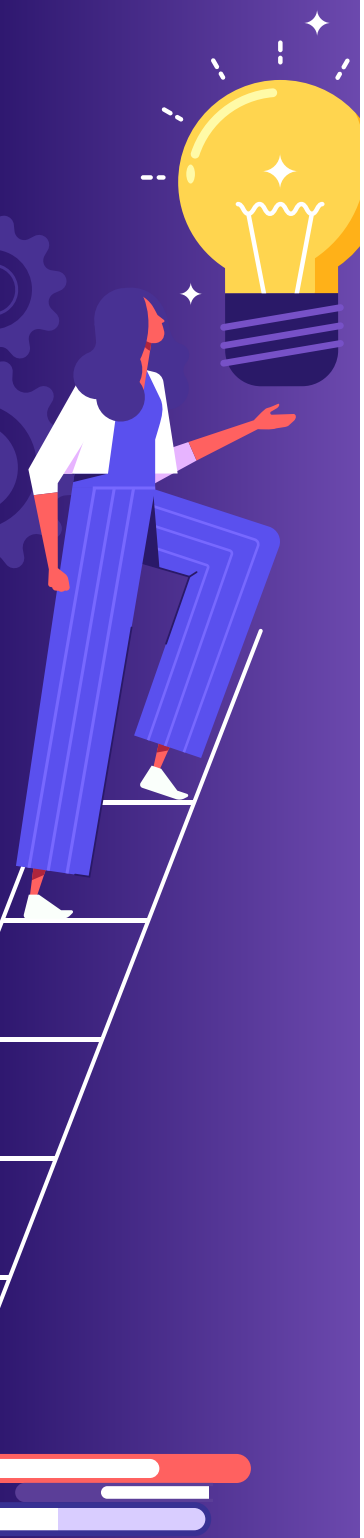
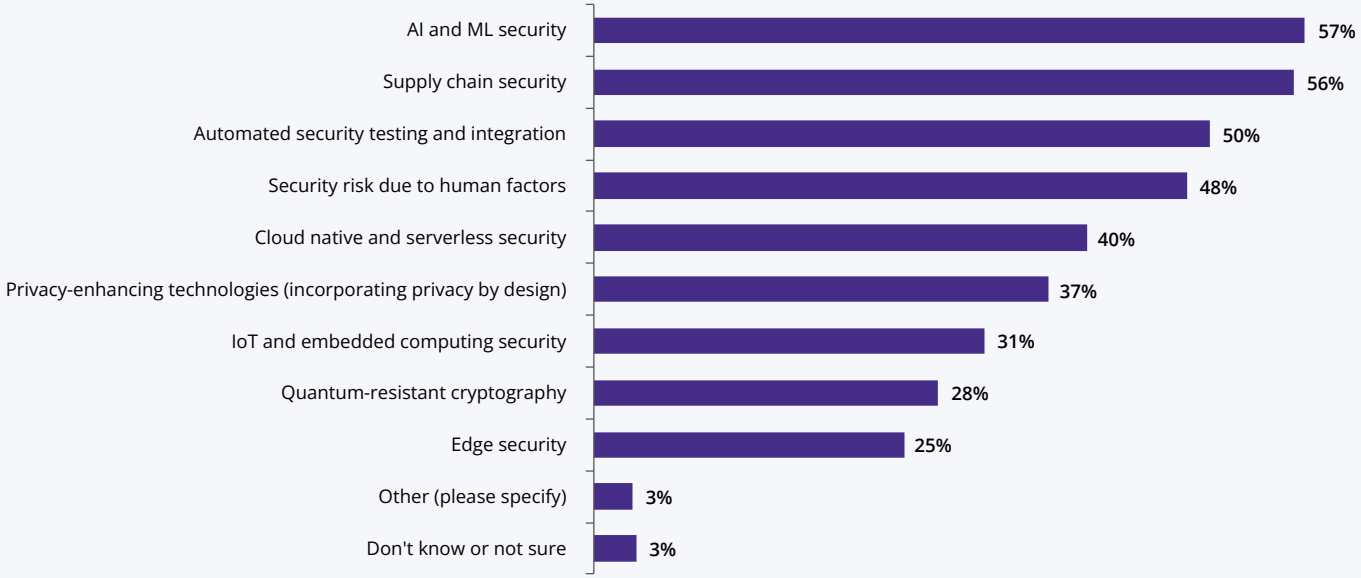


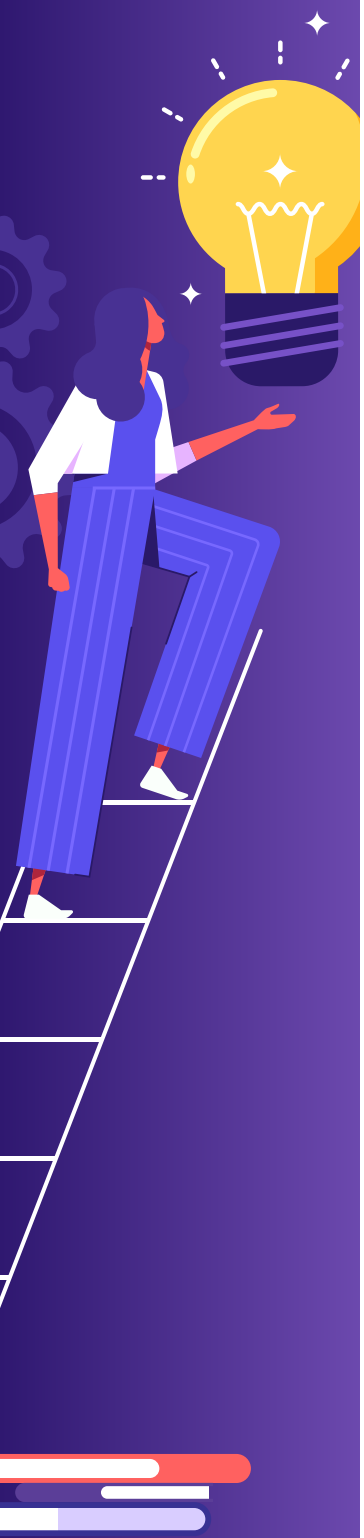
FIGURE 19

AREAS OF SECURE SOFTWARE DEVELOPMENT THAT WILL NEED MORE ATTENTION AND INNOVATION



2024 SecEd Survey, Q29, Sample Size = 324, Total Mentions = 1,227

The survey also highlighted several other areas of concern. Automated testing and integration were noted by 50% of respondents, emphasizing the need for robust mechanisms to continuously identify and address vulnerabilities in an automated manner. Nearly half (48%) mentioned security risks due to human factors, confirming the critical role of human error in cybersecurity breaches. Cloud native and serverless security concerns were raised by 40% of participants, reflecting the shift toward these modern computing paradigms and their unique security demands. Privacy-enhancing technologies were a priority for 37%, evidencing the growing importance of protecting personal data amidst increasing privacy regulations. Other areas of concern were Internet of Things (IoT and embedded computing security (31%), quantum-resistant cryptography (28%), and edge computing (25%).



Chapter 4: OpenSSF course selection

One key reason for having this OpenSSF Security Education survey was to identify what course the OpenSSF should develop next. While the OpenSSF could guess or ask just a few people, it wanted to make decisions based on quantitative data from a widespread survey. Given this survey data, the OpenSSF selected security architecture, as explained below.

Before conducting this survey, the OpenSSF suspected that respondents would generally prefer a language-specific course in a language such as C, Java, or Python. It's true that 54% did indicate language-specific courses as "very important" or "important," indicating that many are interested in such material. However, an even larger 79% indicated that courses not specific to a language were "very important" or "important." This suggests that the OpenSSF should currently focus on creating courses that are not specific to a programming ecosystem.

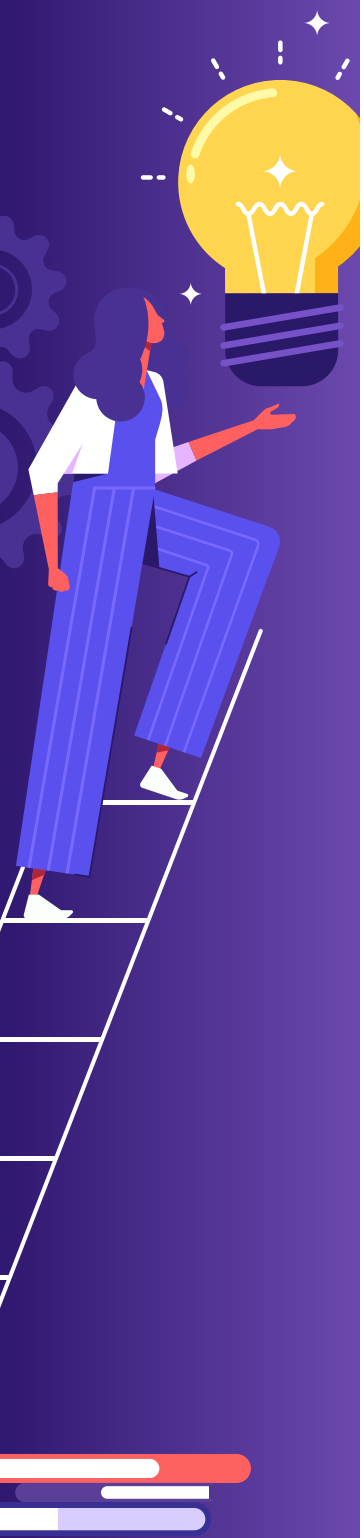
Exactly which area is much more complex because different areas were identified as a "top" choice by different measures:

- In popularity, security architecture and security education and guidance were the most popular, followed by secure implementation, security testing, and threat assessment.
- Considering only first choices, security education and guidance was the most popular choice, followed by strategy and metrics and security architecture.
- In the average ranking when considering popularity, security education and guidance is on top, followed by security architecture and secure implementation.

This variation makes decision-making more complicated. An analysis by roles helps explain why there is such variation. In short, different roles tend to emphasize different areas. Security architecture is the top choice for software developers and system operators and the third top spot for seecurity teams. However, management, data science, and OSPO roles have different priorities. Thus, it shouldn't be surprising that there are multiple "top" answers.

Splitting things up by region revealed an interesting variation: Security architecture was the top choice everywhere except the U.S. and Canada. In the U.S. and Canada, security education and guidance and secure implementation were the top two spots. This suggests that there was a larger mixture of different respondent roles in the U.S. and Canada, leading to more variation.

Years of experience did have a considerable impact. Those with less than five years of experience emphasized security testing as their top pick, while those with five to 20 years of experience emphasized security architecture as their top pick. Those with more than 20 years of experience had security implementation as their top pick with security architecture as their second choice. We have a hypothesis: Less-experienced developers expect that security testing will find all or most of the defects. As developers gain experience, they learn that while such approaches are important, these techniques' false positives and false negatives mean that security architecture has an outsized impact on security, and they want to learn more about that. By the time they have 20 years of experience, they have learned security architecture, and while they are still very interested, keeping up their knowledge in security implementation takes precedence.



While the OpenSSF would love to create all of these courses, it has limited resources and must pick where to start. Some areas seem promising at first but are less so on further consideration:

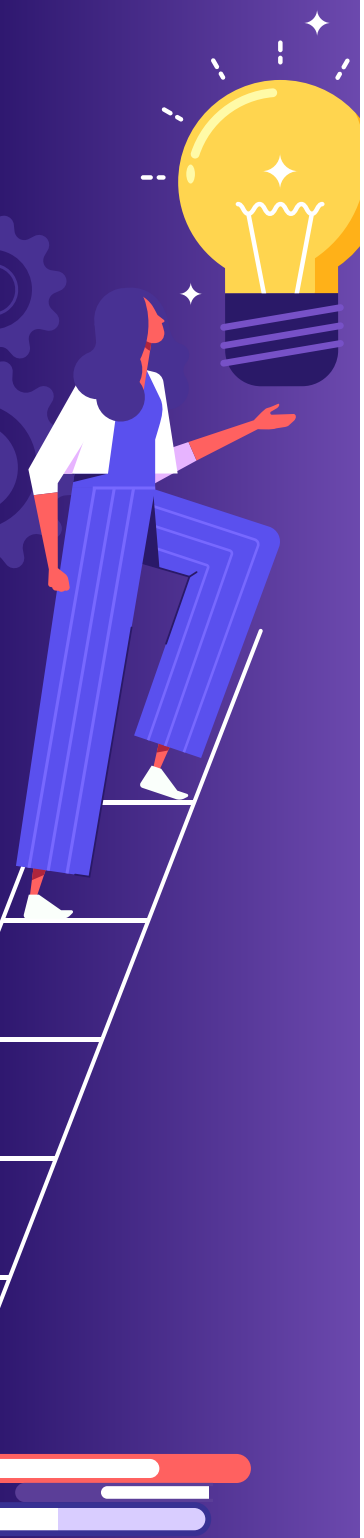
1. Security education and guidance is ranked highly, but it's something of a meta-category. Its definition focused on creating educational systems, yet novices ranked it highly (Figure 33), suggesting that many respondents were not looking at the provided definition but were responding to the notion of wanting more education and guidance in general.
2. Secure implementation is also ranked highly, but the existing OpenSSF fundamentals course already covers secure implementation in a language-independent way. The OpenSSF could go into more depth in secure implementation, but this would essentially require language-specific courses. The other answers indicated that programming language-specific material is important to many but not the most important. Had programming language-specific material ranked much more highly, focusing on this topic would have been appropriate, but with the other answers, it seems less important.
3. Security testing is ranked highly in unstructured feedback and among novices. This is tempting as a decision. Security testing might be a great course after the next one the OpenSSF does. However, it's much less highly ranked among those with experience, and while it ranks highly in some areas, it ranks much lower in others. The OpenSSF should definitely consider creating this course afterward, but it seems less promising as the next course to create.

Those with management roles had different priorities than those with other roles. However, the OpenSSF is already working on a course focused on management. The OpenSSF thinks that a course focused on management would best address their priorities, so for its "201 course," it can focus on others' priorities instead.

At this time, the OpenSSF is planning to focus on security architecture. It's the top area in overall popularity as well as the top choice by software developers and system operators for gap-filling. It also often scores quite highly even in the areas where it isn't the top spot. Many indicated that threat assessment was important, and the OpenSSF could consider including that in a security architecture course. No one topic is the top choice for everyone, but given the trade-offs, this appears to be a good choice.

There are some courses in security architecture but not many, and most only discuss a short list of principles. The OpenSSF's current fundamentals course does discuss security architecture, but like other courses, it mostly discusses a short list of principles. As a result, a security architecture course could be a clear follow-on course that easily extends the existing material.

Thus, the OpenSSF believes that addressing security architecture next would best meet respondents' needs.



Chapter 5: About the survey and its respondents

This study is based on a web survey conducted by the Linux Foundation and its partners from March 1 through April 19, 2024. We received 398 valid responses, 318 of which completed the whole survey. Moreover, some questions were not intended for all respondents, as noted in Table 2, which describes the structure of the survey. Therefore, the sample size for the different analyses can vary, as noted in the captions of the figures throughout this report.

In the following, we present the demographics of the respondents and the study methodology. The full survey instrument is available at **http://www.data.world/thelinuxfoundation**.

Demographics

Figure 20 presents the demographics of the respondent organizations. In terms of organization size based on the number of employees, we classified respondents into small (1–249), medium (250–999), and large (20,000+) organizations. A similar number of respondents from each organization size participated in the survey; 31% were small, 35% were medium, and 33% were large. In terms of type of company, there is a balance between organizations that consume IT products and services but operate in other areas (47%) and organizations whose revenue stream comes primarily from IT products and services (41%). There are also other types

TABLE 2

STRUCTURE OF THE SURVEY

Pages	Questions	Question categories	Who answers the questions
P1		Introduction	All respondents
P2–P3	Q1–Q6	Tell us about yourself	All respondents (N=398)
P4	Q7–Q8	Involvement in open source	Open source contributors (N=270)
P5	Q9–Q13	Tell us about the organization that you work for	Only employed professionals (N=362)
P6	Q14–Q21	Perspectives on secure software development	All respondents (N=398)
P7–P9	Q22–Q30	Educational needs for secure software development	All respondents (N=322–352)
P9	Q31–Q32	Reasons for non-use of education materials	Respondents who have not taken courses (N=135–150)
P10	Q33	LFR Panel and reward information	All respondents

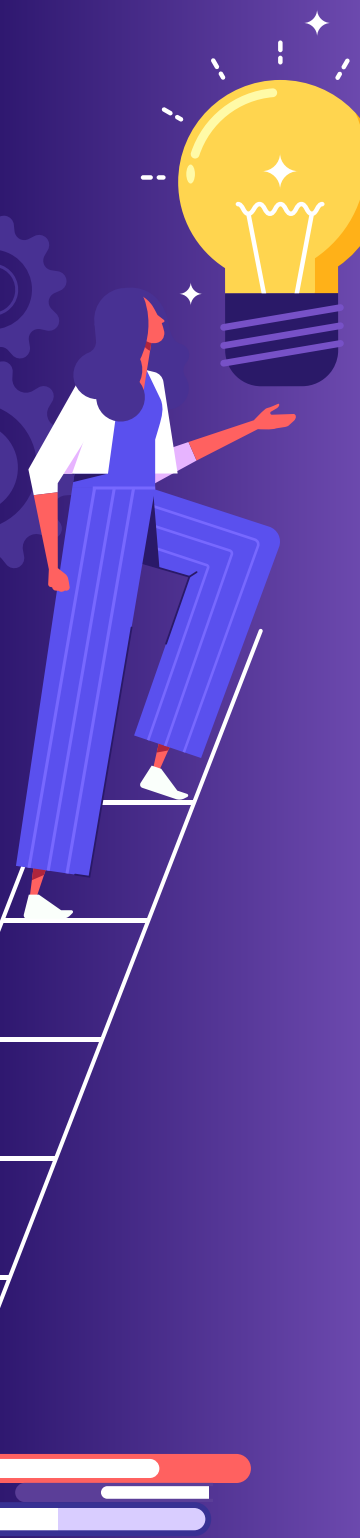
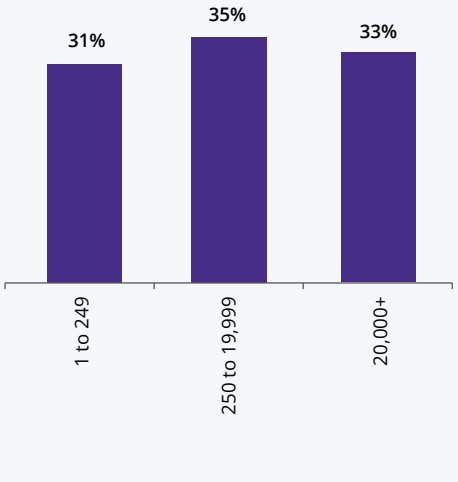


FIGURE 20

ORGANIZATIONAL DEMOGRAPHICS

Organization size

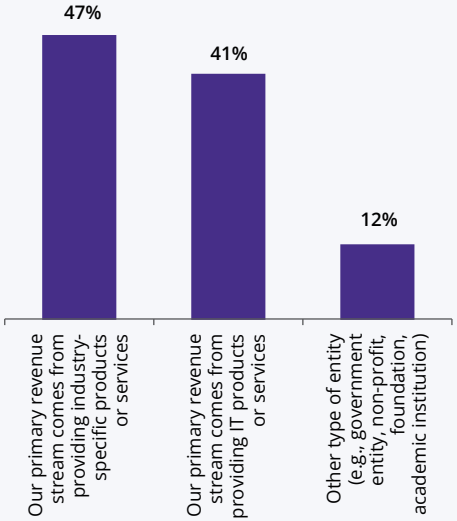
Please estimate how many total employees are in the company or entity you work for.



2024 SecEd Survey, Q12, Sample Size = 356, DKNS excluded, result of the regrouping of other answers

Type of company

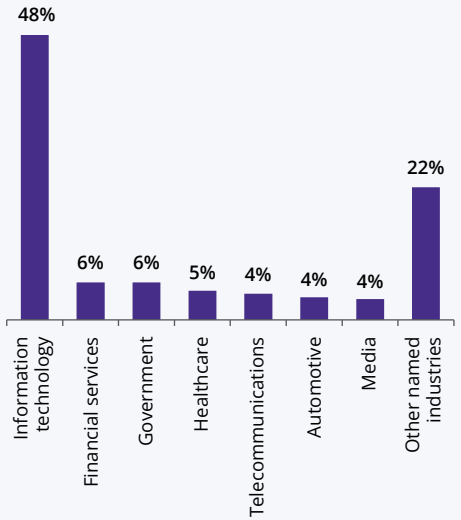
Which type of company or entity do you work for?



2024 SecEd Survey, Q10, Sample Size = 362

Industry

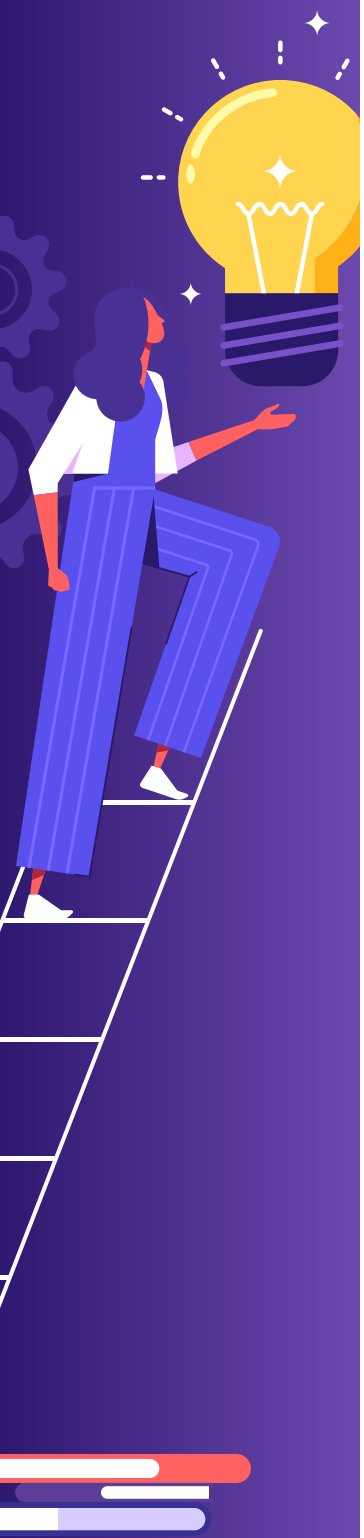
Which of the following best describes your company's or entity's primary industry?



2024 SecEd Survey, Q11, Sample Size = 362

of organizations (12%), such as government entities, non-profits, foundations, and academic institutions, represented in the survey. The panel on the right provides a window into the organization's primary industry. Overall, information technology (IT vendor, service provider, or manufacturer) accounts for 48% and other industries for 52% of the sample. The strong showing of IT is not surprising, given the survey's focus. Other named industries, totaling 22%, were retail, education, utilities, transportation, and others (totaling less than 3% in the sample).

Figure 21 shows some demographics of the respondents. Software developers comprise half of the sample (50%), followed by security team (16%), management (12%), system operations (9%), and others. Most respondents are employed full time (82%) and represent multiple perspectives, with some being able to speak for themselves (37%), the department (28%), the company (20%), or the industry (16%). Most participants are involved in OSS to some extent, with 17% contributing less than one hour per week and 51% contributing more. The most common role is occasional contributor (39%), followed by maintainer (28%),



non-development contributor (14%), core contributor (9%), and committer (8%). Regarding geographic region, the majority of respondents are located in Europe (41%) or the U.S. or Canada (36%), while 13% come from the Asia-Pacific region. The remaining 9% are from other parts of the world. The survey data reveals a diverse range of experience levels, with 20% having less than five years, 53% having five to 20 years, and the remaining 27% having more than 20 years of experience in software development. However, when focusing on secure software development, 30% have less than two years, 48% have three to 10 years, and 22% have more than 10 years of experience. Finally, in terms of familiarity, nearly half of the respondents (47%) consider themselves very or extremely familiar with secure software development, 28% report being not familiar or slightly familiar, and 25% report being just familiar.

Methodology and open results data

The study data was collected via an online survey promoted via social media, the Linux Foundation and Linux.com websites, and the Linux Foundation Newsletter and with the support of the OpenSSF. We received 786 responses, but 388 were discarded for not meeting the screening criteria or passing the quality checks. The screening criteria for participants included being involved in software application development, confirming their human status in a question designed to filter out bots, and being able to speak about the topic. The quality check involved ensuring sufficient data for analysis, which was measured by the number of questions answered and the frequency of "Don't know or not sure" (DKNS) responses. Additionally, the quality check encompassed a thorough manual review of open-ended responses, the time spent on the survey, and patterns in the answers provided. The final sample size analyzed for the survey was 398. To access the survey dataset, see **http://www.data.world/thelinuxfoundation**.

It is worth noting that participation in the survey was voluntary, which may introduce self-selection bias. This type of bias occurs when participants choose to be part of a study or survey based on characteristics that also influence the outcome of interest, potentially skewing the results.

How missing data is handled. Although respondents are required to answer nearly all questions in the survey (the only exceptions are some open-ended questions), there are times when a respondent is unable to answer a question because it is outside the scope of their role or experience. For this reason, we frequently add a DKNS response to the list of responses for a question. However, this creates a conundrum regarding what to do with DKNS responses. One approach is to treat it just like any other response. In this way, report readers can see the percentage of respondents that answered DKNS. The advantage of this approach is that it reports back the exact distribution of the data collected. The challenge with this approach is that it distorts the distribution of valid responses—those responses where respondents could answer the question.

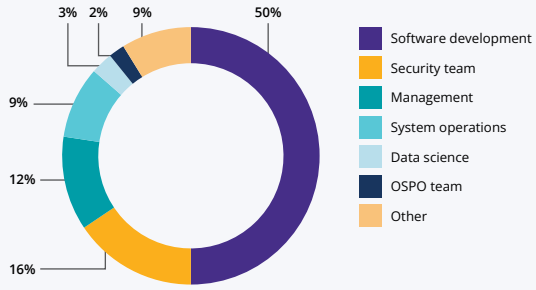
Some of the analyses in this report excluded the DKNS. This can be done because the data missing can be classified as either missing at random or missing completely at random. Excluding DKNS data from a question does not change the distribution of data (counts) for the other responses, but it does change the size of the denominator used to calculate the percent of responses across the remaining responses. This has the effect of proportionally increasing the percent values of the remaining responses relative to the number of DKNS responses. The number of valid cases is adjusted accordingly. Where we have elected to exclude DKNS data, a careful examination of the footnote for the figure will enable the reader to determine the number of DKNS responses based on the difference between the sample size (DKNS included) and valid cases (DKNS excluded). Finally, percentage values in this report may not add up to exactly 100% due to rounding.

FIGURE 21

RESPONDENT DEMOGRAPHICS

Role

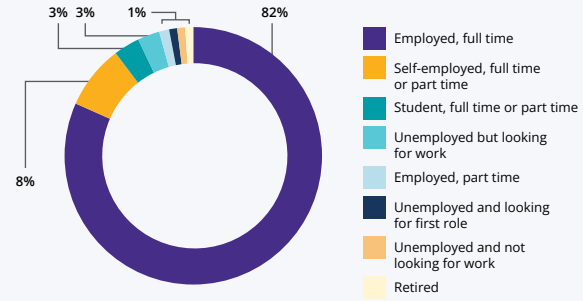
Professionally, which role or field do you most closely identify with?



Sample Size = 398, simpler names for the categories and merging IT development – Director or Vice President, IT operations – Director or Vice President, Product or project management, and C-level under the Management categories. All the others under Other

Employment

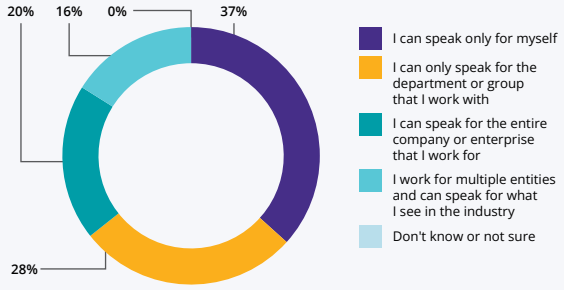
What is your current employment status?



2024 SecEd Survey, Q3, Sample Size = 398

Perspective

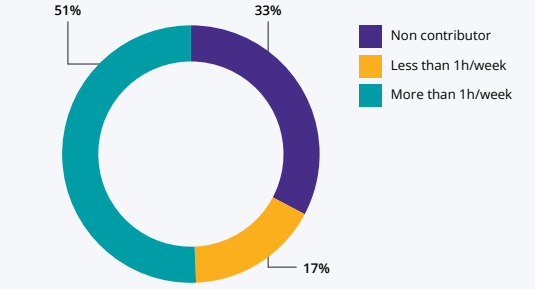
What perspective will you speak for in this survey?



2024 SecEd Survey, Q4, Sample Size = 398

Contribution to OSS

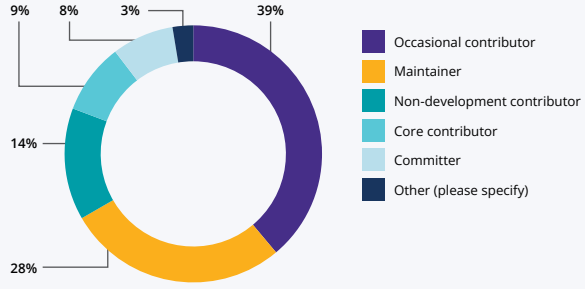
How many hours do you contribute to open source projects per week on average?



2024 SecEd Survey, Combined Q6 & Q7, Sample Size = 391, DNKS excluded

OSS role

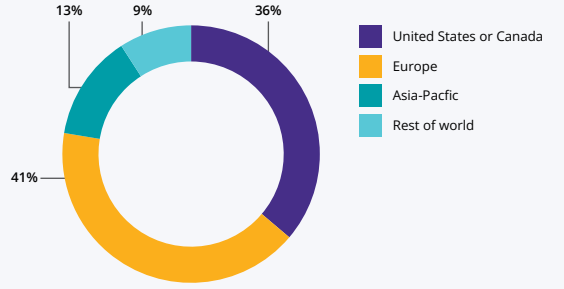
What role occupies most of your time in the open source projects you are involved with?



2024 SecEd Survey, Q8, Sample Size = 270

Region

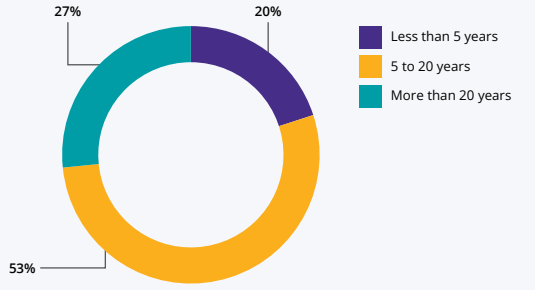
In what country or region are you based?



2024 SecEd Survey, Q9, Sample Size = 362, Asia-Pacific = China, India, Japan, Oceania, Asia Pacific (except...). Rest of World = Mexico, Central America, the Caribbean, or South America, Middle East, Other (please specify)

Experience in software development

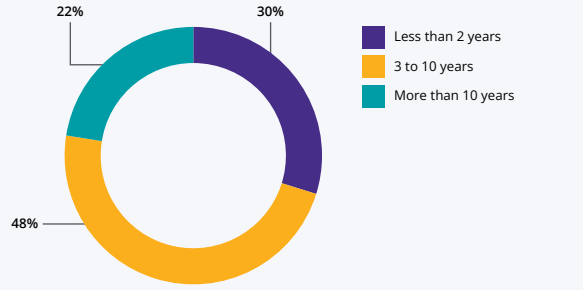
How many years of experience do you have in software development?



2024 SecEd Survey, Q15, Sample Size = 395, DKNS excluded, regrouping of more specific answers

Experience in secure software development

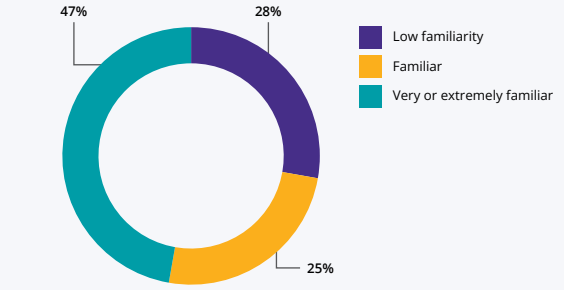
How many years of experience do you specifically have in secure software development?



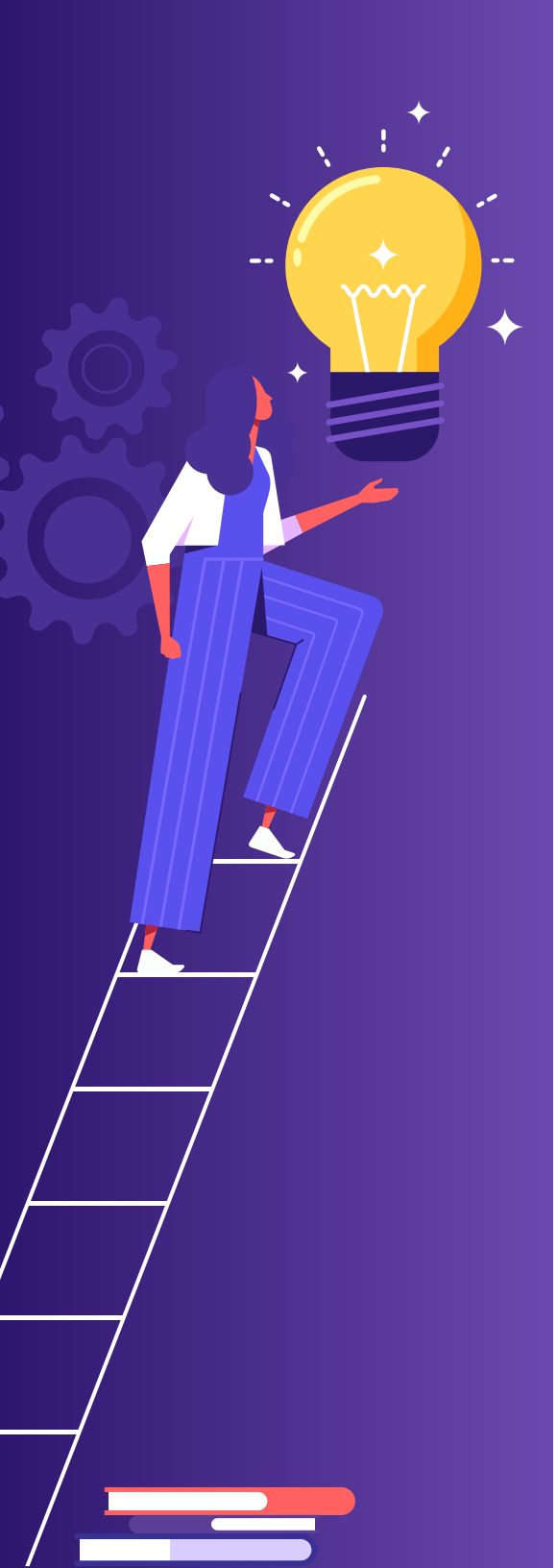
2024 SecEd Survey, Q16, Sample Size = 369, DKNS excluded, regrouping of more specific answers

Familiarity

How familiar are you with secure software development?



2024 SecEd Survey, Q14, Sample Size = 396, DKNS excluded.
Low familiarity = Not familiar at all + Somewhat familiar
Very or extremely familiarity = Very familiar + Extremely familiar



Conclusion

In conclusion, our survey has revealed significant gaps in the current state of secure software development knowledge and training among professionals. A substantial portion of developers, including those with extensive experience, lack familiarity with secure development practices. Most professionals rely on on-the-job experience as a main learning resource, but it takes many years of such experience to achieve a minimum level of familiarity. New coursework materials can accelerate this process and remove the major challenge for implementing secure software development, as pointed out by the survey respondents.

The findings from our survey highlight the importance of language-agnostic courses, particularly in areas such as security architecture, security education and guidance, and secure implementation. Furthermore, there is a clear demand for Python-specific training, reflecting the language's widespread use and critical role in the software ecosystem. However, training needs vary significantly based on professional roles and experience levels, evidencing the need for diverse educational offerings in secure software practices.

The OpenSSF's decision to introduce a new course on security architecture is a step in the right direction, addressing one of the most popular and critical areas identified in the survey. The OpenSSF is also taking steps to increase awareness of the current OpenSSF educational materials, e.g., by including references to them in other Linux Foundation newsletters and materials.

By making all the survey data openly available, we encourage further exploration and use of these insights to foster a culture of "security by design" in software development education. Ensuring that developers are equipped with the necessary skills and knowledge to implement secure software development effectively will be instrumental in building resilient systems that protect sensitive data and maintain user trust.

Appendix A: Cybersecurity in the organizations

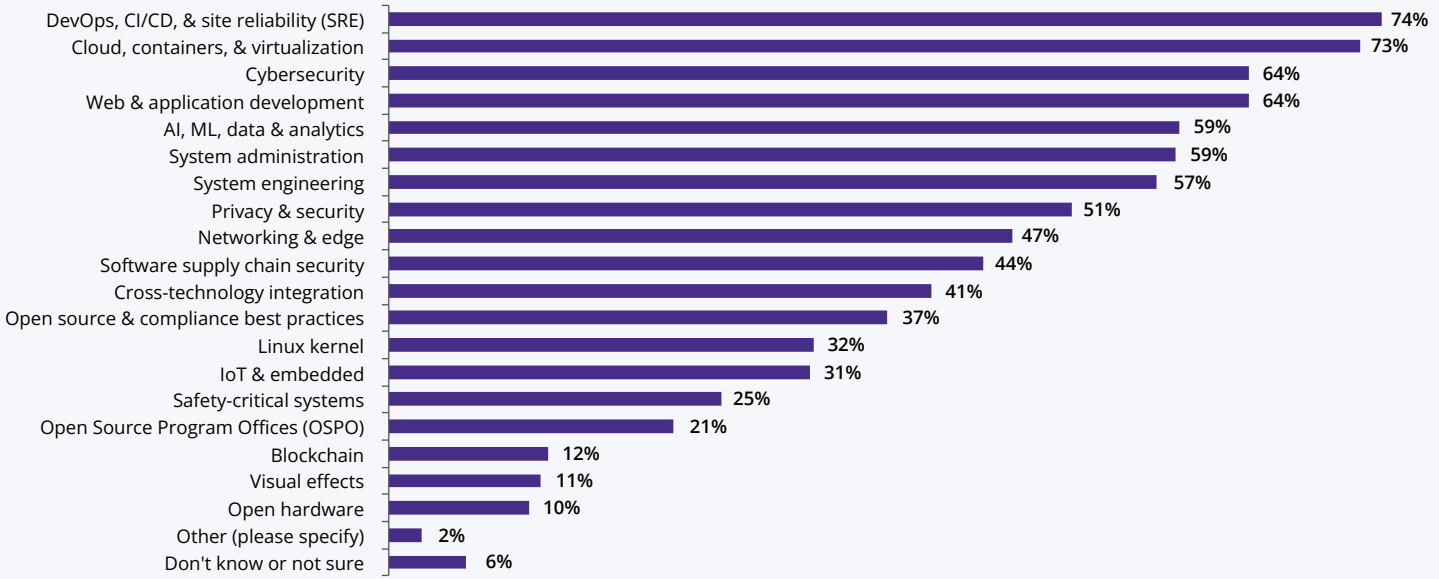
In addition to the analysis specific to the need for training in secure software development and the priority areas, we explored topics related to cybersecurity in the organizations in the survey. This appendix discusses the technical headcount in this area, the activities adopted by organizations, and the resources to stay up to date on the latest security vulnerabilities or threats.

Cybersecurity is a priority for organizations

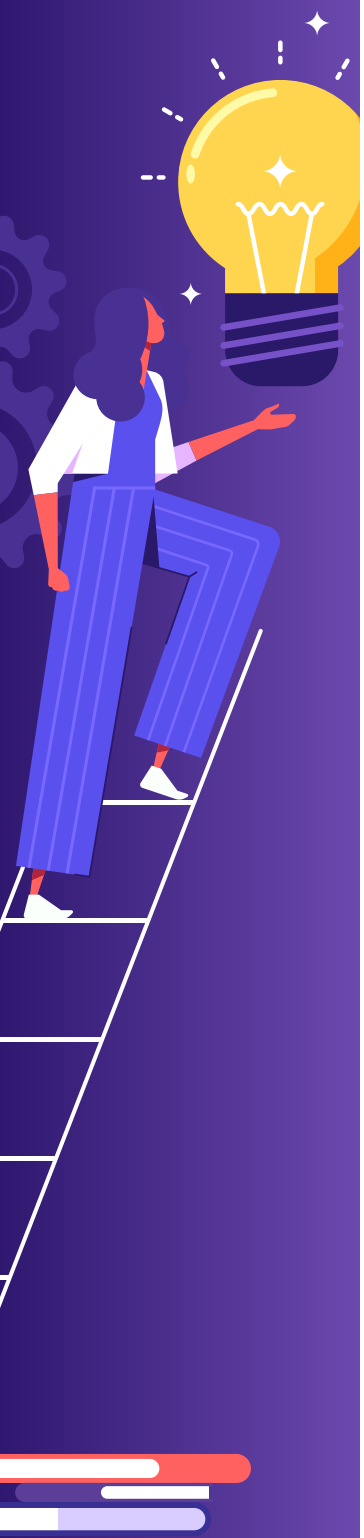
Cybersecurity is a priority for most organizations, with 64% of them staffing this area with technical headcount, making it the third most common area, as shown in Figure 22. Figure 23 shows that cybersecurity is also a priority for IT end-user organizations, with 64% reporting staff in this area compared with 65% in IT providers. Even among smaller organizations, cybersecurity remains crucial; 51% of those with fewer than 250 employees

FIGURE 22

TECHNOLOGICAL AREAS STAFFED BY TECHNICAL HEADCOUNT IN RESPONDENT ORGANIZATIONS



2024 SecEd Survey, Q13, Sample Size = 362, Total Mentions = 2,978



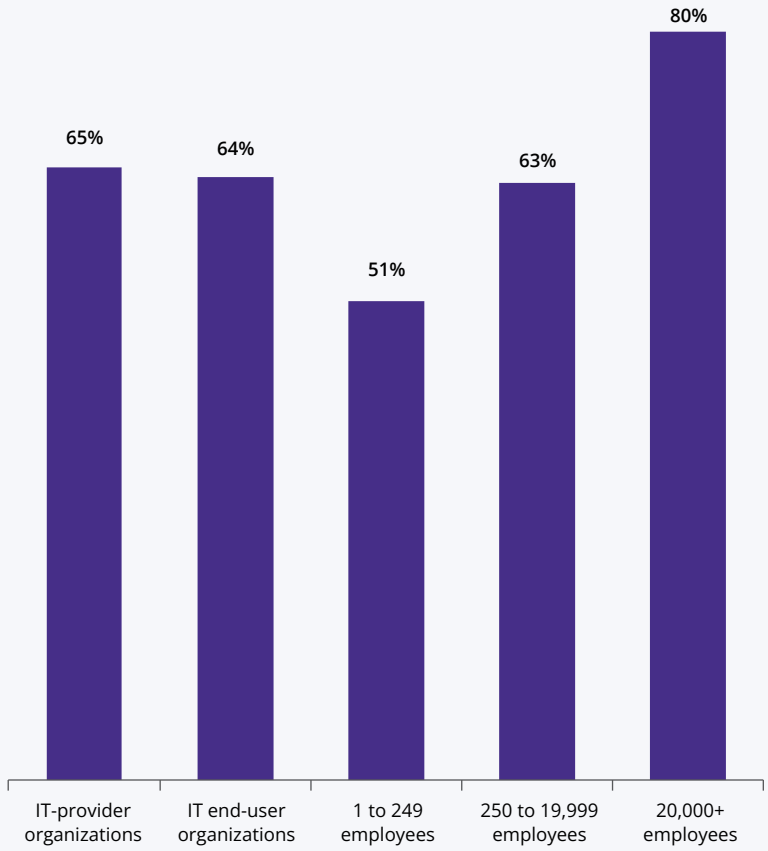
have dedicated staff in this area. The emphasis on cybersecurity escalates with organizational size: sixty-three percent of organizations with 250 to 19,999 employees report headcount in cybersecurity, rising to 80% among organizations with over 20,000 employees.

This high level of investment in cybersecurity personnel reflects the growing awareness and urgency to protect digital assets against an ever-evolving landscape of cyberthreats. Today, attackers from around the world can threaten organizations through cyberattacks. The necessity for cybersecurity experts is also driven by regulatory requirements and compliance standards, which in some cases mandate strict data protection protocols. Ensuring compliance with these standards not only safeguards sensitive information but also contributes to the organization's reputation and trustworthiness.

However, while staffing cybersecurity professionals is crucial, it is not sufficient on its own; cybersecurity needs to be deeply integrated into the software development process. Security should be considered throughout software development, including design, implementation, verification, and deployment. By embedding security practices and principles into the development lifecycle, organizations can proactively identify and address vulnerabilities, reduce the risk of breaches, and create more resilient software. For an effective integration of security aspects in the software development lifecycle processes, software professionals must be familiar with the techniques and technologies of secure software development.

FIGURE 23

PERCENTAGE OF RESPONDENTS THAT REPORT STAFF HEADCOUNT IN CYBERSECURITY



2024 SecEd Survey, Q13 vs. Q10, Q13 vs. Q12, Sample Size = 362

Organizations adopt a variety of cybersecurity activities

Figure 24 depicts the cybersecurity activities incorporated into organizations' software development and deployment processes. CI or CI/CD, when considered as a combined option, is the most widely adopted practice, with 75% of respondents including it in their workflows. This high adoption rate highlights the opportunity to integrate secure software development tools and practices not only for building this infrastructure but also for checking code before it goes into production. Logging (68%), secret management (67%), and monitoring & alerting (66%) are also prominently featured, highlighting the emphasis on tracking and responding to security incidents in real time.

Most organizations implement unit testing (66%), indicating a strong focus on validating code integrity. Most organizations also implement identity and access management (65%), showing a widespread desire to manage user permissions. Configuration management, security patching, and secure design & implementation of software are other critical activities, each cited by more than 60% of respondents. On the other hand, activities such as fuzz testing (26%) and cyberthreat intelligence (28%) are less commonly included, suggesting potential areas for further improvement and investment.

Online courses are an important resource for organizations

Staying up to date with the latest security vulnerabilities and threats is important for organizations to safeguard their digital assets and maintain operational integrity. Proactively updating and patching systems can prevent potential breaches that could lead to substantial financial losses and damage to reputation. Cybercrime was estimated to cost organizations $8.15 trillion (USD) in 2023, and that number is expected to rise.[12] Moreover, in some fields, regulatory compliance requires strict adherence to security practices, making continuous vigilance a necessity rather than an option. It's true that most vulnerabilities are of the same kinds of vulnerabilities as in decades past. For example, in 2023, 75% of exploited zero-days in important, widely used software were memory safety vulnerabilities,[13,14] a problem originally identified and discussed in the 1970s. However, new vulnerabilities in specific products are regularly and need to be promptly addressed, even though they are often the same types of vulnerabilities. In addition, new types of vulnerabilities (or ways to more easily exploit them) are occasionally discovered, such as

Cybercrime was estimated to cost organizations $8.15 trillion (USD) in 2023, and that number is expected to rise. Moreover, in some fields, regulatory compliance requires strict adherence to security practices, making continuous vigilance a necessity rather than an option.

12 **https://www.statista.com/chart/28878/expected-cost-of-cybercrime-until-2027**

13 **https://googleprojectzero.blogspot.com/2022/04/the-more-you-know-more-you-know-you.html**

14 75% (42/56) of the 2023 entries in **https://docs.google.com/spreadsheets/d/1lkNJ0uQwbeC1ZTRrxdtuPLCIl7mlUreoKfSIgajnSyY/edit**

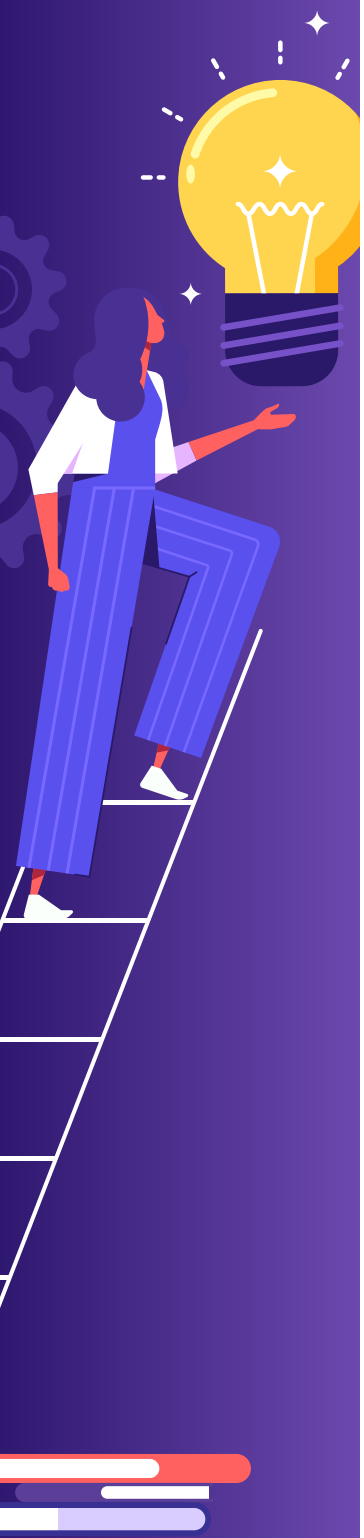
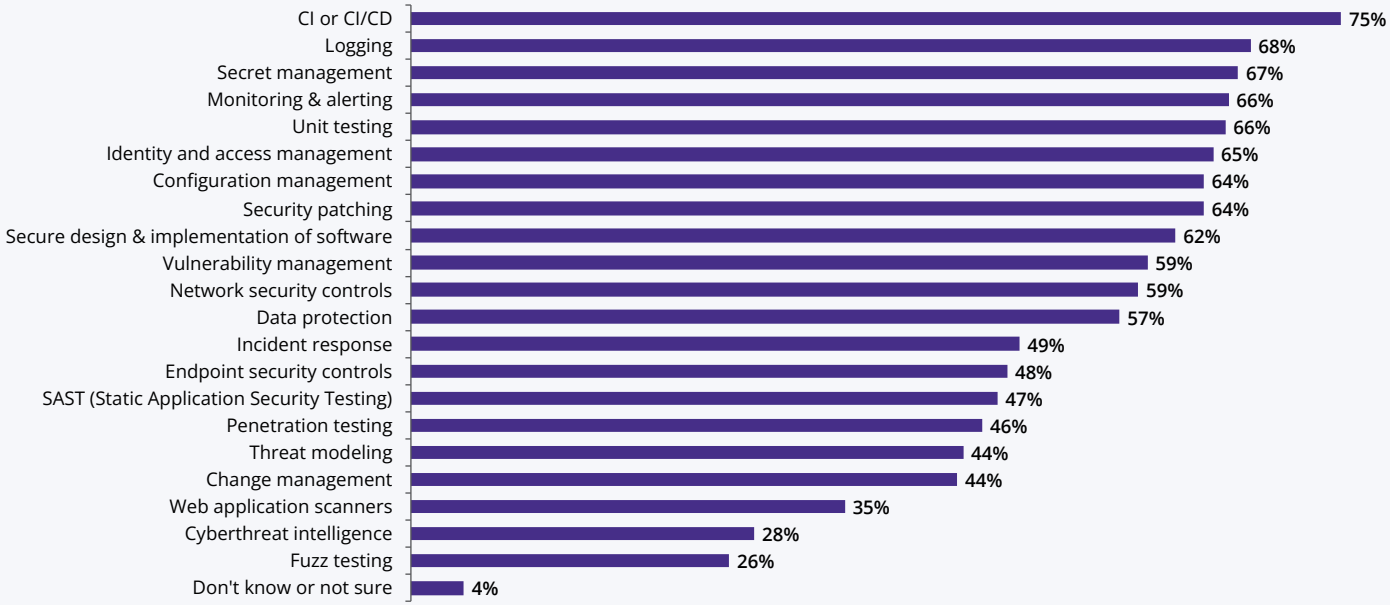


FIGURE 24

CYBERSECURITY ACTIVITIES ADOPTED BY ORGANIZATIONS AS PART OF THEIR SOFTWARE DEVELOPMENT AND DEPLOYMENT PROCESSES

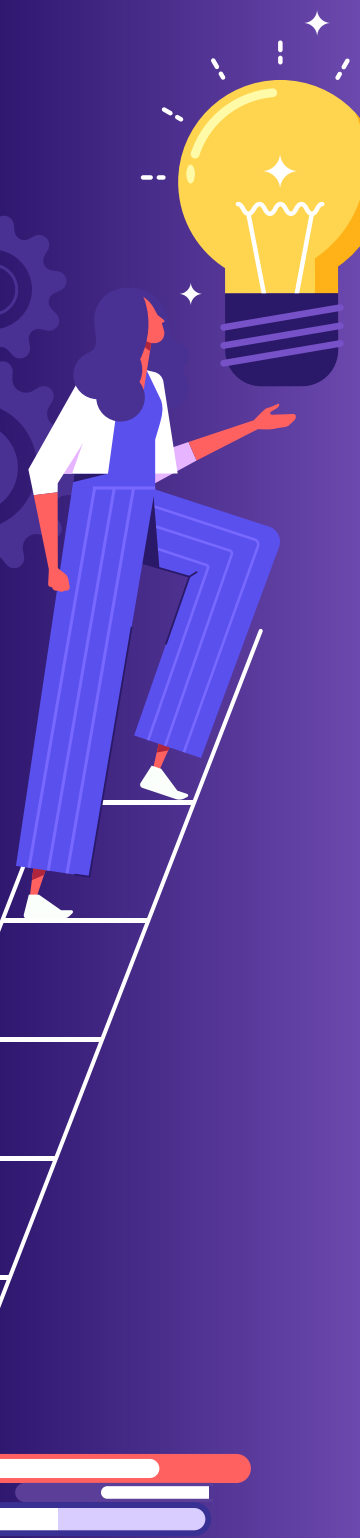


2024 SecEd Survey, Q18, Sample Size = 398, Total Mentions = 4,538

the discovery of dependency confusion attacks in 2021, leading to changes in how to best counter them.[15] Thus, organizations should stay informed and responsive to new security challenges.

Even though security websites, databases, blogs, and mailing lists are unsurprisingly the top resources used for receiving the latest information, continuous learning and certification are quite popular among respondents. As observed in Figure 25, 40% of the organizations leverage this resource to stay tuned about the latest developments in the area. By encouraging employees to engage in ongoing education and achieve professional certifications, organizations can help their teams keep their cybersecurity knowledge and skills up to date. This proactive approach enables personnel to reduce their organizations' risks as well as identify and respond to new vulnerabilities more effectively. As a result, continuous learning and certification not only enhance an

15 https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610

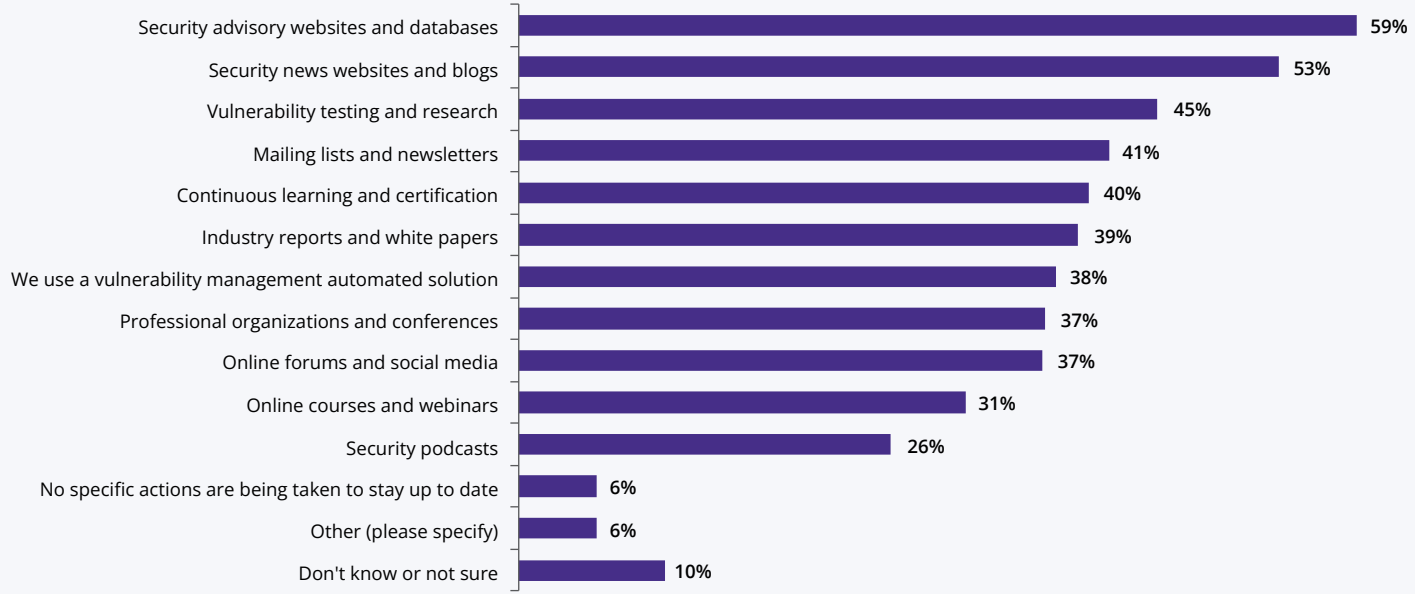


organization's security posture but also foster a culture of security awareness and preparedness, which is crucial for mitigating risks in this evolving landscape.

The relevance of continuous learning and certification changes across different organizational sectors, as observed in Figure 26: The OSPO team shows the highest engagement, with 63% of respondents acknowledging the importance of these educational resources. Following closely are the security team (56%) and management (55%), indicating a strong recognition of continuous learning's value in these critical areas. Other roles, however, report less reliance on such resources, with system operations at 36% and software development at 31%. Figure 26 also reveals that larger organizations are more likely to adopt these educational resources. Specifically, 57% of organizations with 20,000 or more employees report using continuous learning and certification as key resources. This figure drops to 40% for organizations with employee counts ranging from 250 to 19,999 and further decreases to 25% for smaller entities with 1 to 249 employees. This trend highlights a correlation between the size of an organization and its commitment to maintaining up-to-date security measures through ongoing education.

FIGURE 25

RESOURCES FOR STAYING UP TO DATE ON THE LATEST SECURITY VULNERABILITIES OR THREATS RELATED TO TECHNOLOGIES IN USE



2024 SecEd Survey, Q19, Sample Size = 398, Total Mentions = 1,861

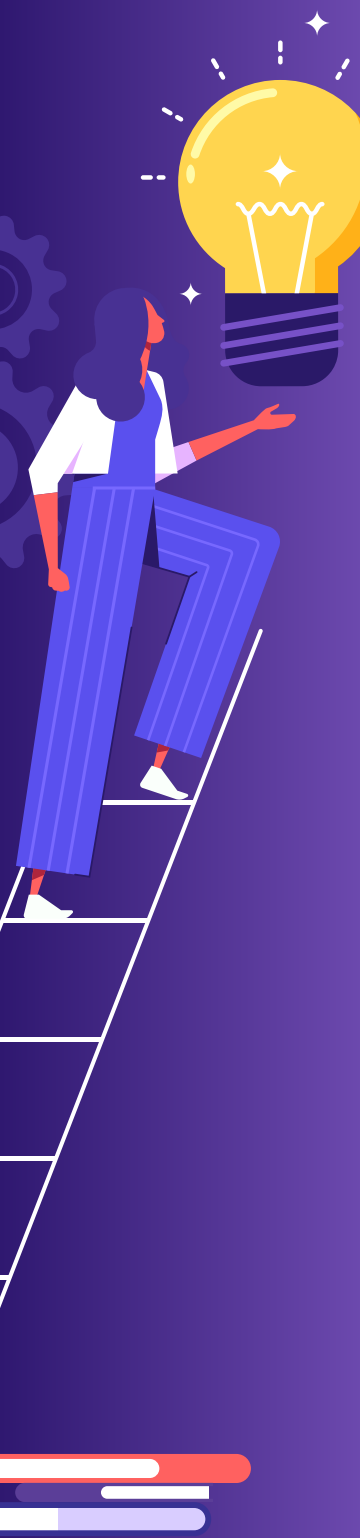
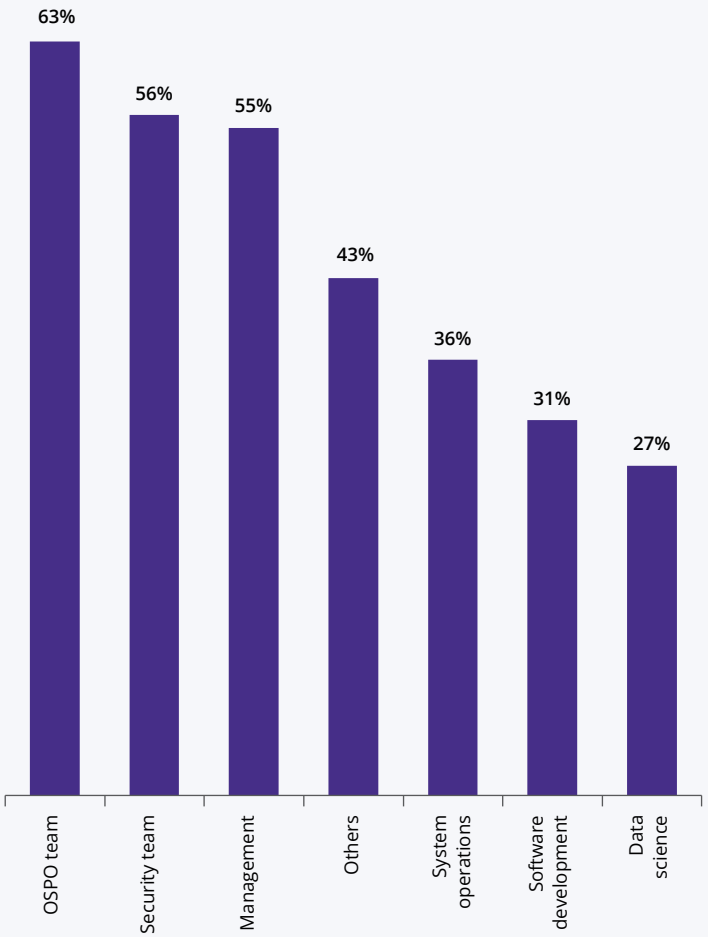


FIGURE 26

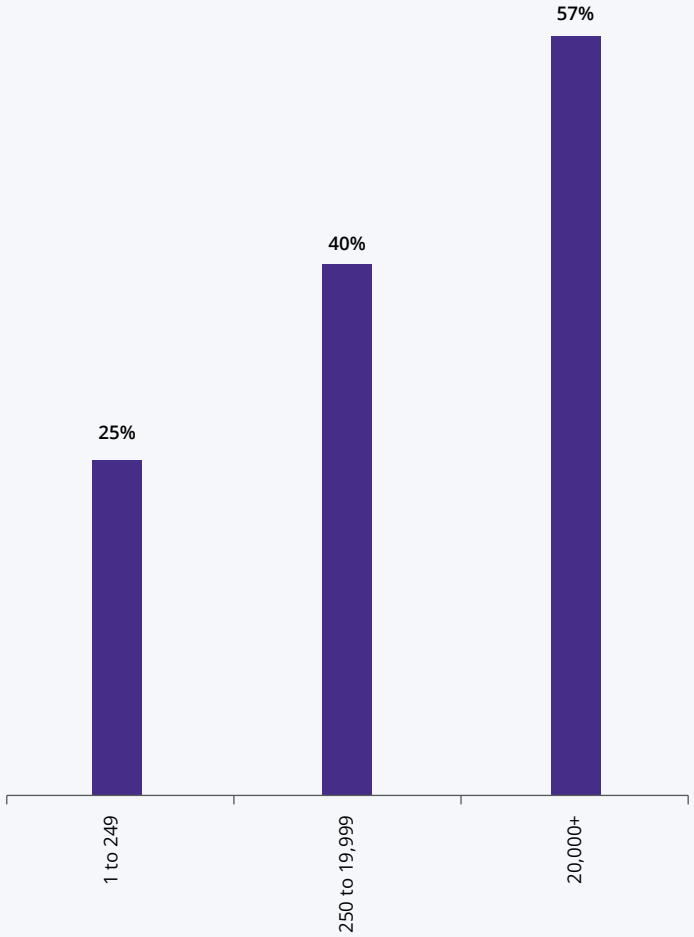
PERCENTAGE OF RESPONDENTS THAT REPORT CONTINUOUS LEARNING AND CERTIFICATION AS A RESOURCE FOR STAYING UP TO DATE ON THE LATEST SECURITY VULNERABILITIES OR THREATS

Segmented by role



2024 SecEd Survey, Q19 by Q5, Sample Size = 398, percentage of those who report "Continuous learning and certification" for the question "How does your organization stay up to date on the latest security vulnerabilities or threats related to the technologies that you use?

Segmented by number of employees in the organization



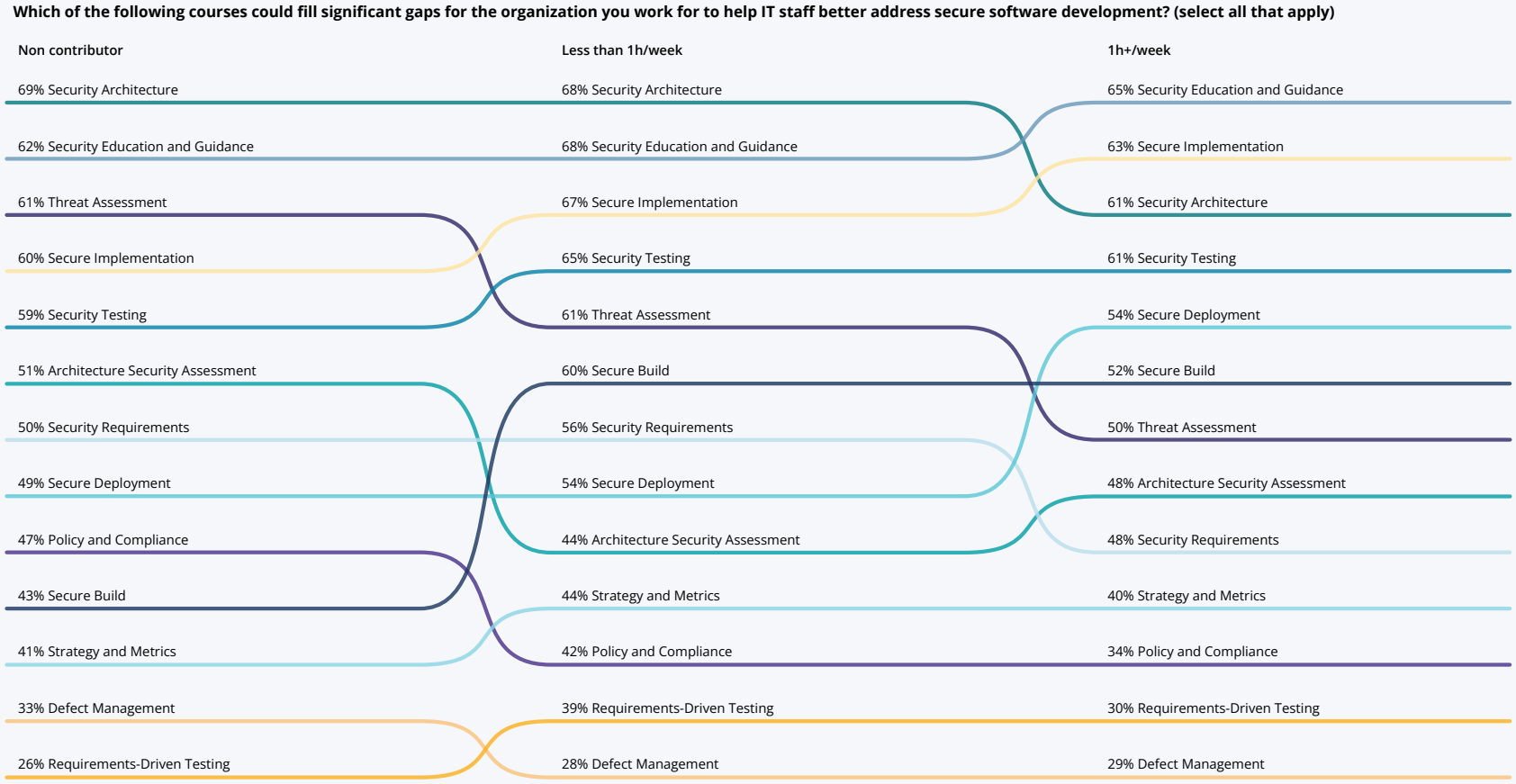
2024 SecEd Survey, Q19 by Q12, Sample Size = 356, percentage of those who report "Continuous learning and certification" for the question "How does your organization stay up to date on the latest security vulnerabilities or threats related to the technologies that you use?

Appendix B: Segregated rankings for language-agnostic courses

The following figures show the rankings of language-agnostic courses segregated by various criteria. Unsurprisingly, the relative importance of different courses varies depending on a variety of factors.

FIGURE 27

LANGUAGE-AGNOSTIC COURSES BY CONTRIBUTION TO OSS



2024 SecEd Survey, Q25 by Q7, Sample Size = 319, Total Mentions = 2,105, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 28

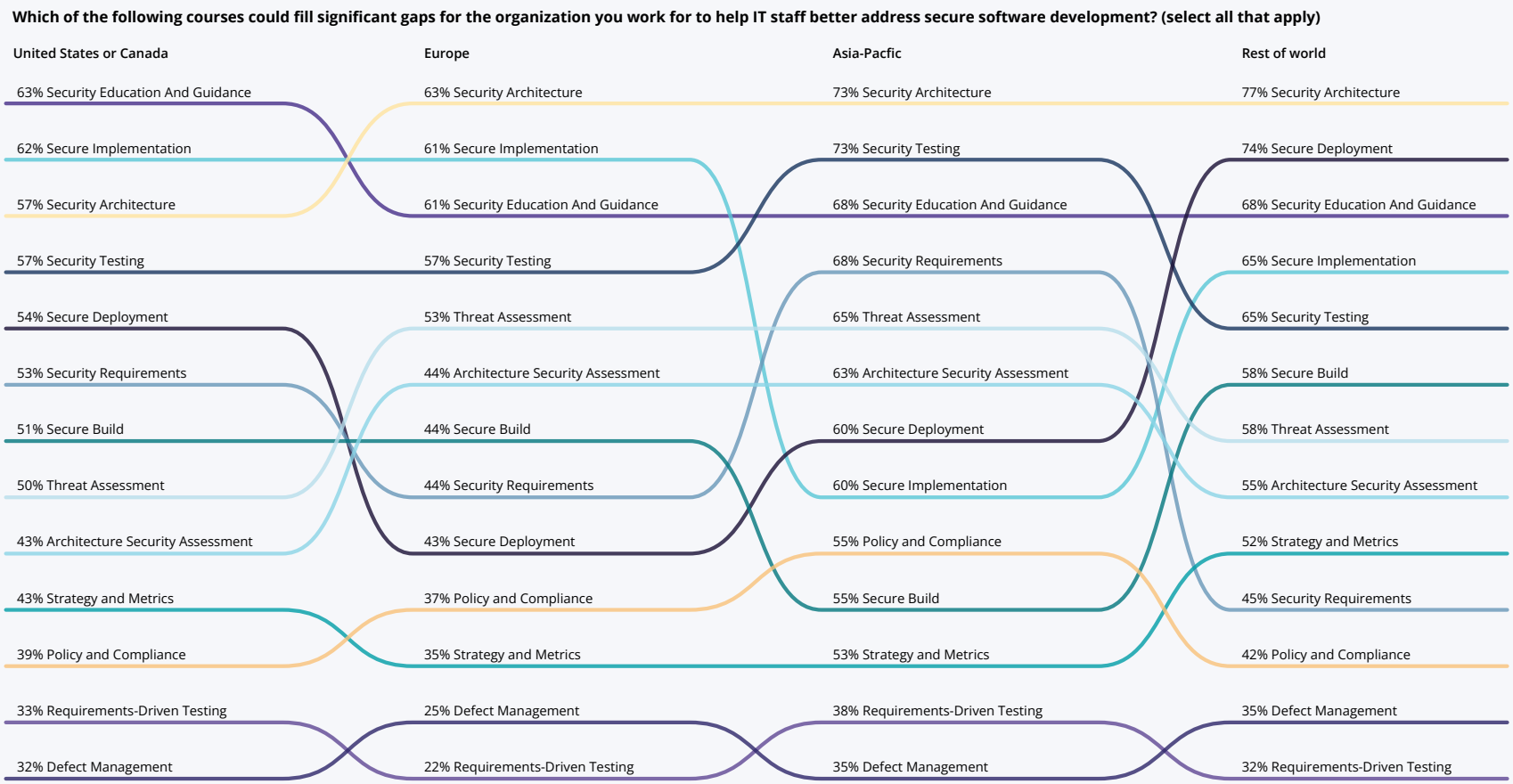
LANGUAGE-AGNOSTIC COURSES BY OSS ROLE

Which of the following courses could fill significant gaps for the organization you work for to help IT staff better address secure software development? (select all that apply)

2024 SecEd Survey, Q25 by Q8, Sample Size = 231, Total Mentions = 1,512, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 29

LANGUAGE-AGNOSTIC COURSES BY REGION

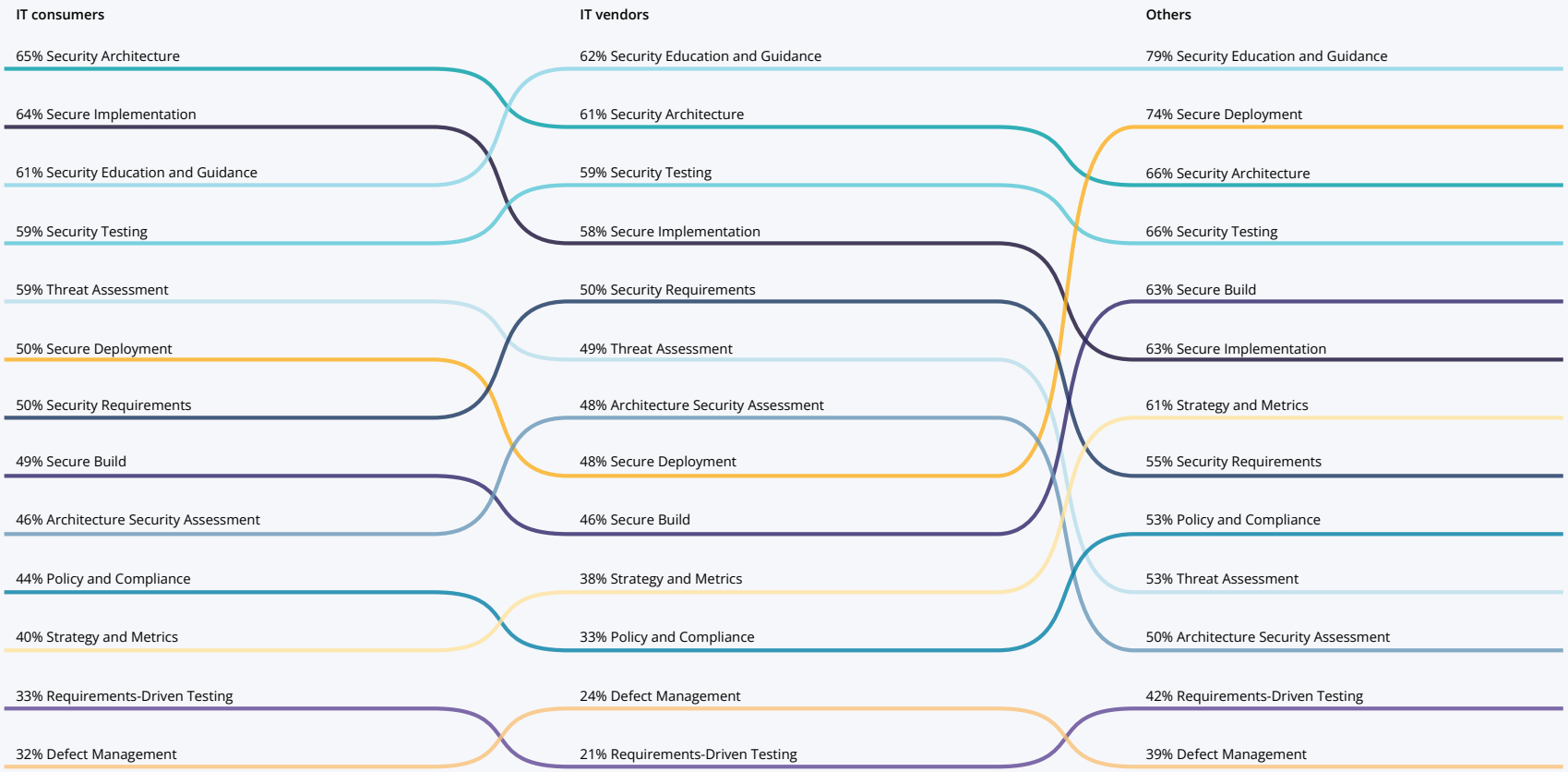


2024 SecEd Survey, Q25 by Q9, Sample Size = 312, Total Mentions = 2,028, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 30

LANGUAGE-AGNOSTIC COURSES BY ORGANIZATION TYPE

Which of the following courses could fill significant gaps for the organization you work for to help IT staff better address secure software development? (select all that apply)

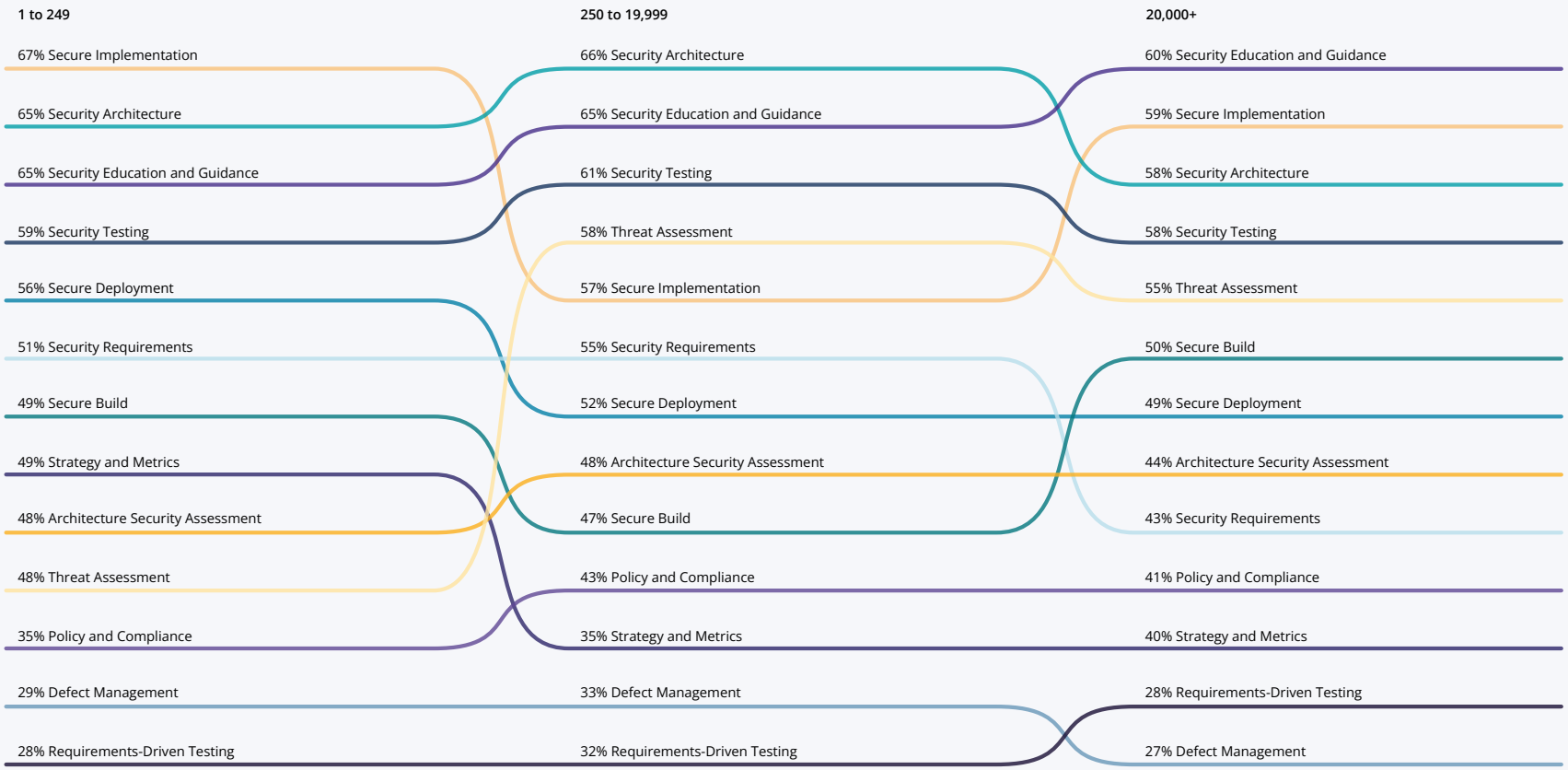


2024 SecEd Survey, Q25 by Q10, Sample Size = 312, Total Mentions = 2,028, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 31

LANGUAGE-AGNOSTIC COURSES BY ORGANIZATION SIZE

Which of the following courses could fill significant gaps for the organization you work for to help IT staff better address secure software development? (select all that apply)

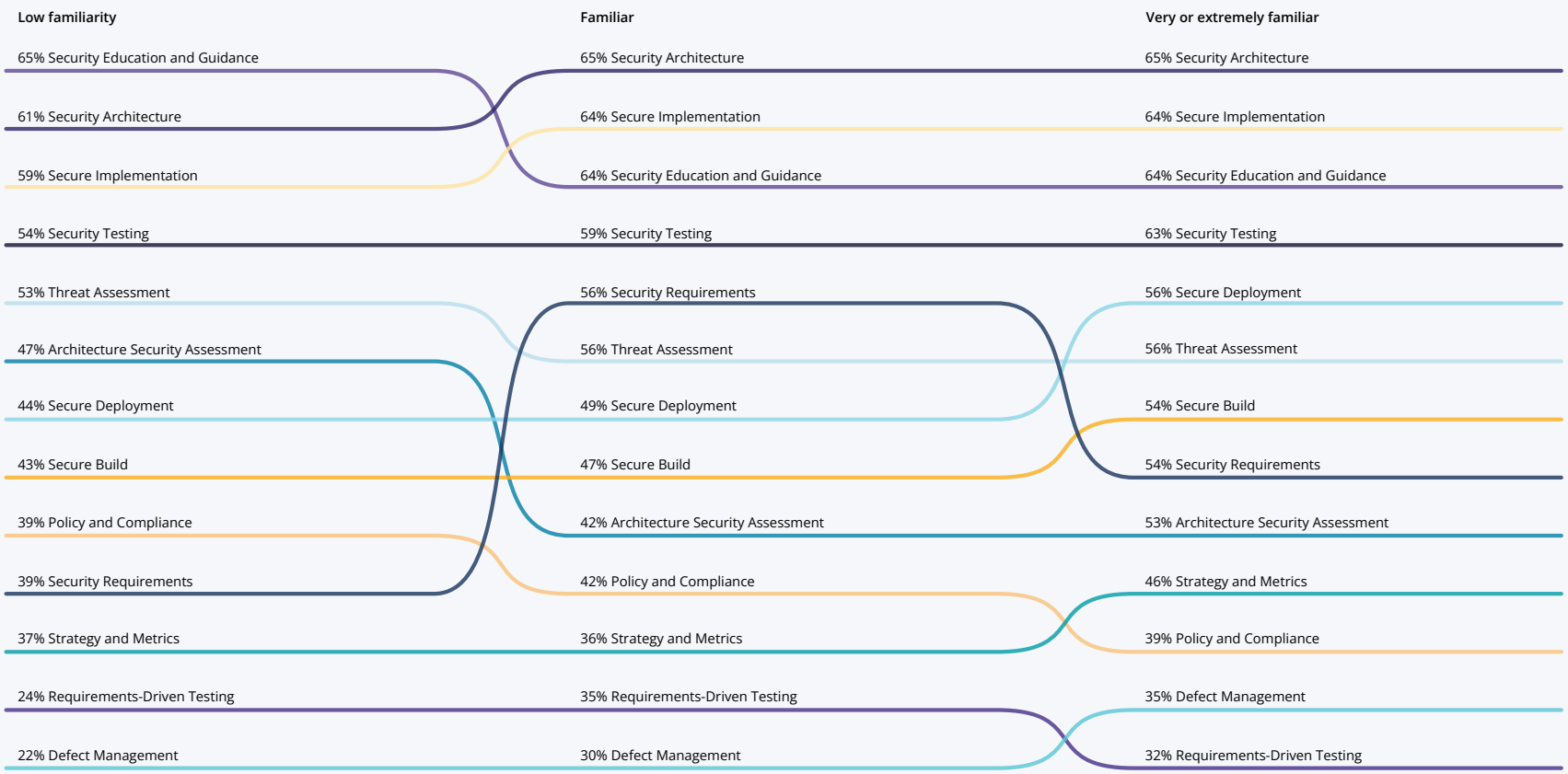


2024 SecEd Survey, Q25 by Q12, Sample Size = 307, Total Mentions = 1,983, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 32

LANGUAGE-AGNOSTIC COURSES BY FAMILIARITY

Which of the following courses could fill significant gaps for the organization you work for to help IT staff better address secure software development? (select all that apply)

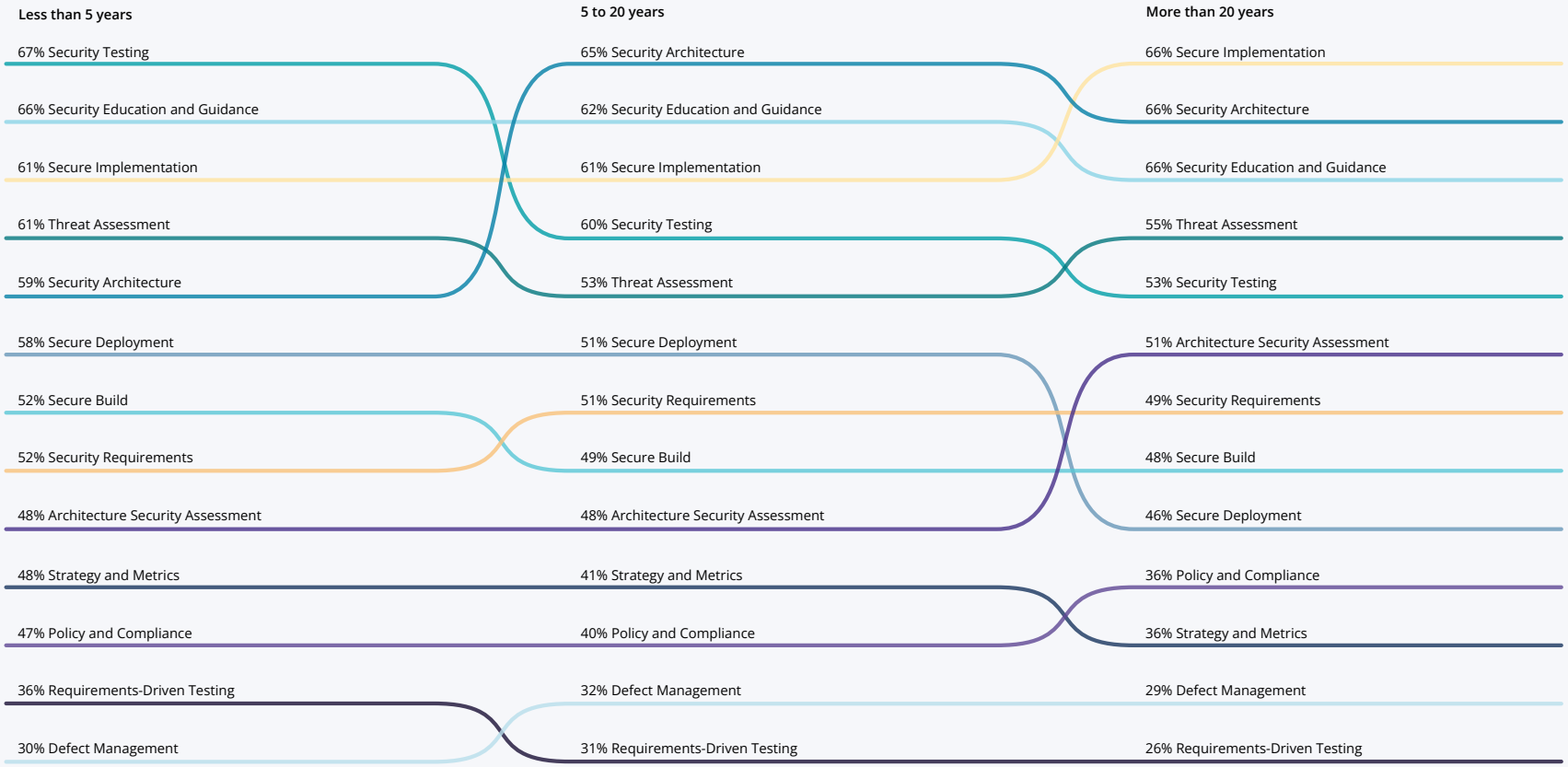


2024 SecEd Survey, Q25 by Q14, Sample Size = 340, Total Mentions = 2,223, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 33

LANGUAGE-AGNOSTIC COURSES BY YEARS OF EXPERIENCE

Which of the following courses could fill significant gaps for the organization you work for to help IT staff better address secure software development? (select all that apply)



2024 SecEd Survey, Q25 by Q15, Sample Size = 340, Total Mentions = 2,227, the number in front of the name represents the percentage of respondents, each column is sorted by this number

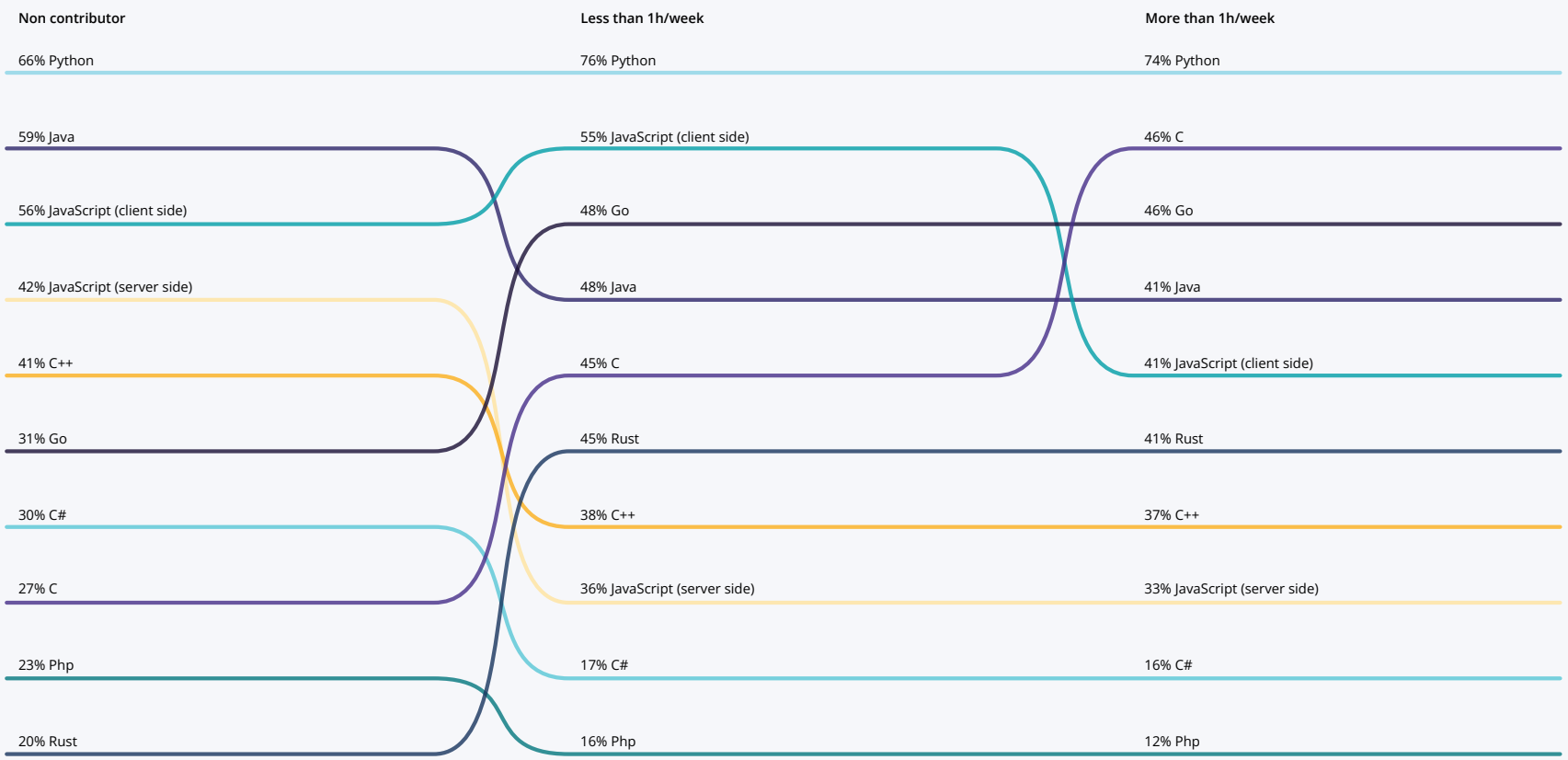
Appendix C: Segregated rankings for language-specific courses

The following figures show the rankings of language-specific courses segregated by various criteria. Unsurprisingly, the relative importance of different language-specific courses varies depending on a variety of factors.

FIGURE 34

LANGUAGE-SPECIFIC COURSES BY CONTRIBUTION TO OSS

Which language-specific ecosystem course(s) on secure software development should the organization you work for make available to its developers? (select all that apply)

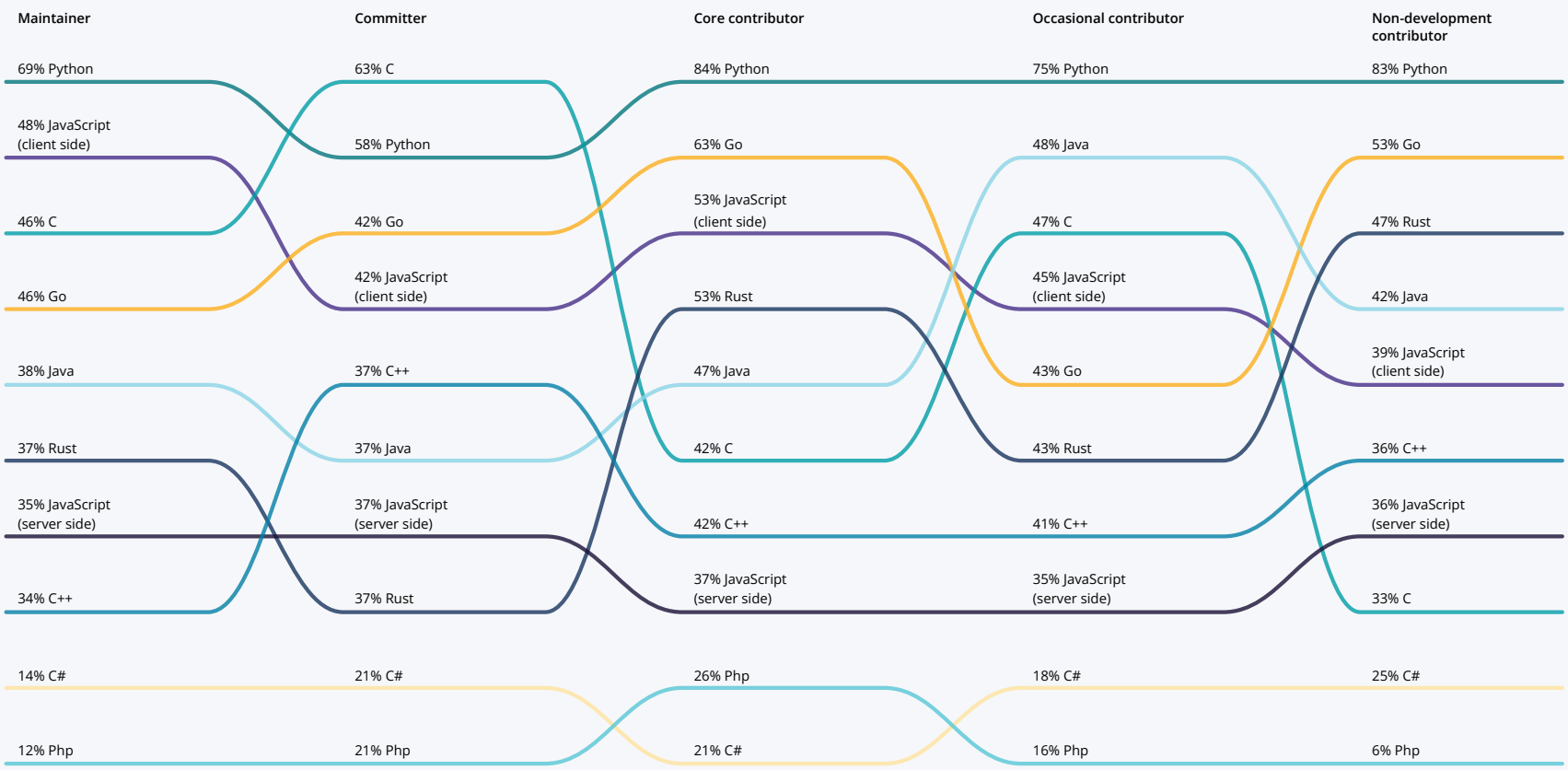


2024 SecEd Survey, Q23 by Q7, Sample Size = 329, Total Mentions = 1,342, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 35

LANGUAGE-SPECIFIC COURSES BY OSS ROLE

Which language-specific ecosystem course(s) on secure software development should the organization you work for make available to its developers? (select all that apply)



2024 SecEd Survey, Q23 by Q8, Sample Size = 239, Total Mentions = 991, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 36

LANGUAGE-SPECIFIC COURSES BY REGION

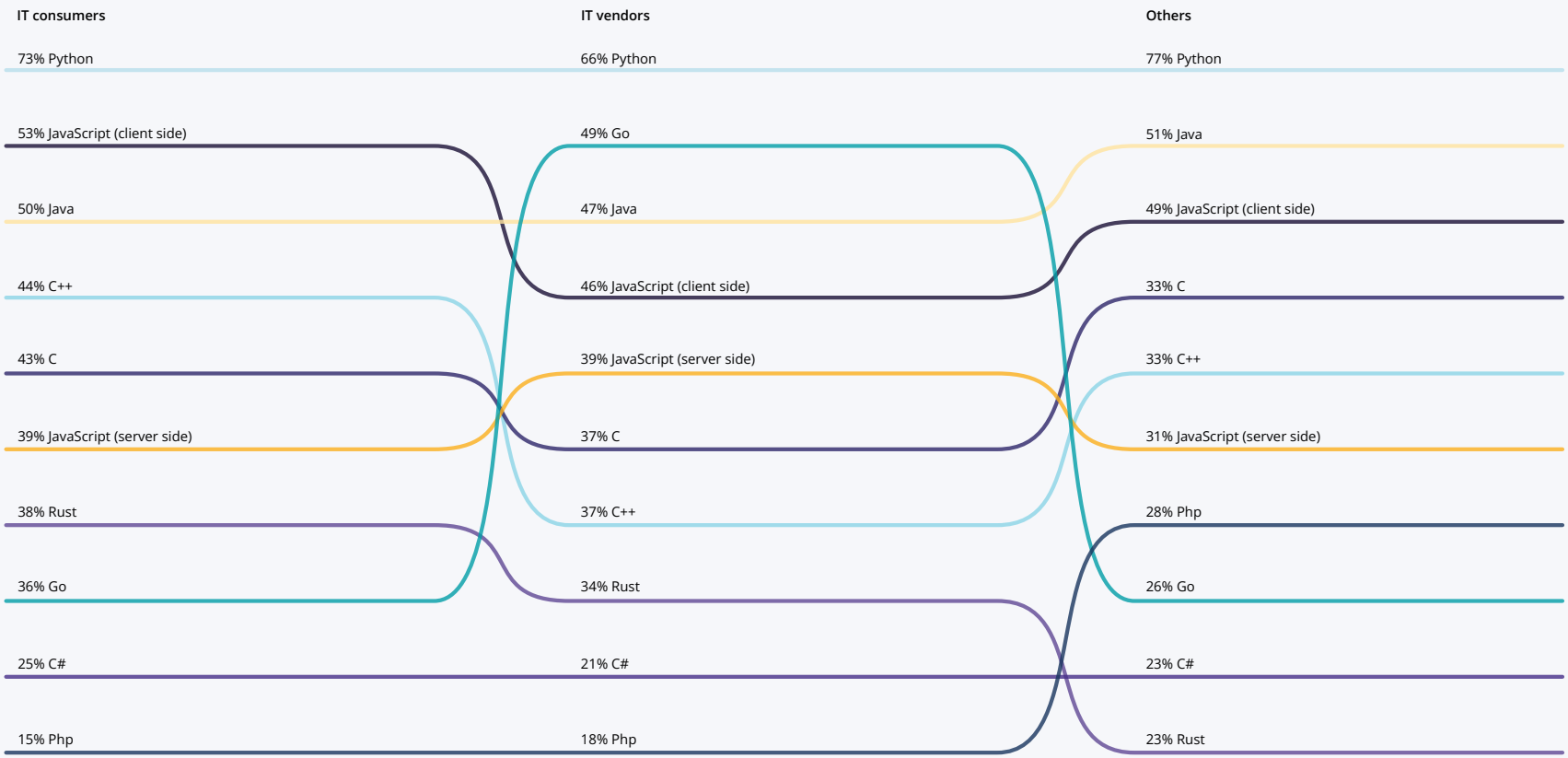
Which language-specific ecosystem course(s) on secure software development should the organization you work for make available to its developers? (select all that apply)

2024 SecEd Survey, Q23 by Q9, Sample Size = 321, Total Mentions = 1,330, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 37

LANGUAGE-SPECIFIC COURSES BY TYPE OF ORGANIZATION

Which language-specific ecosystem course(s) on secure software development should the organization you work for make available to its developers? (select all that apply)

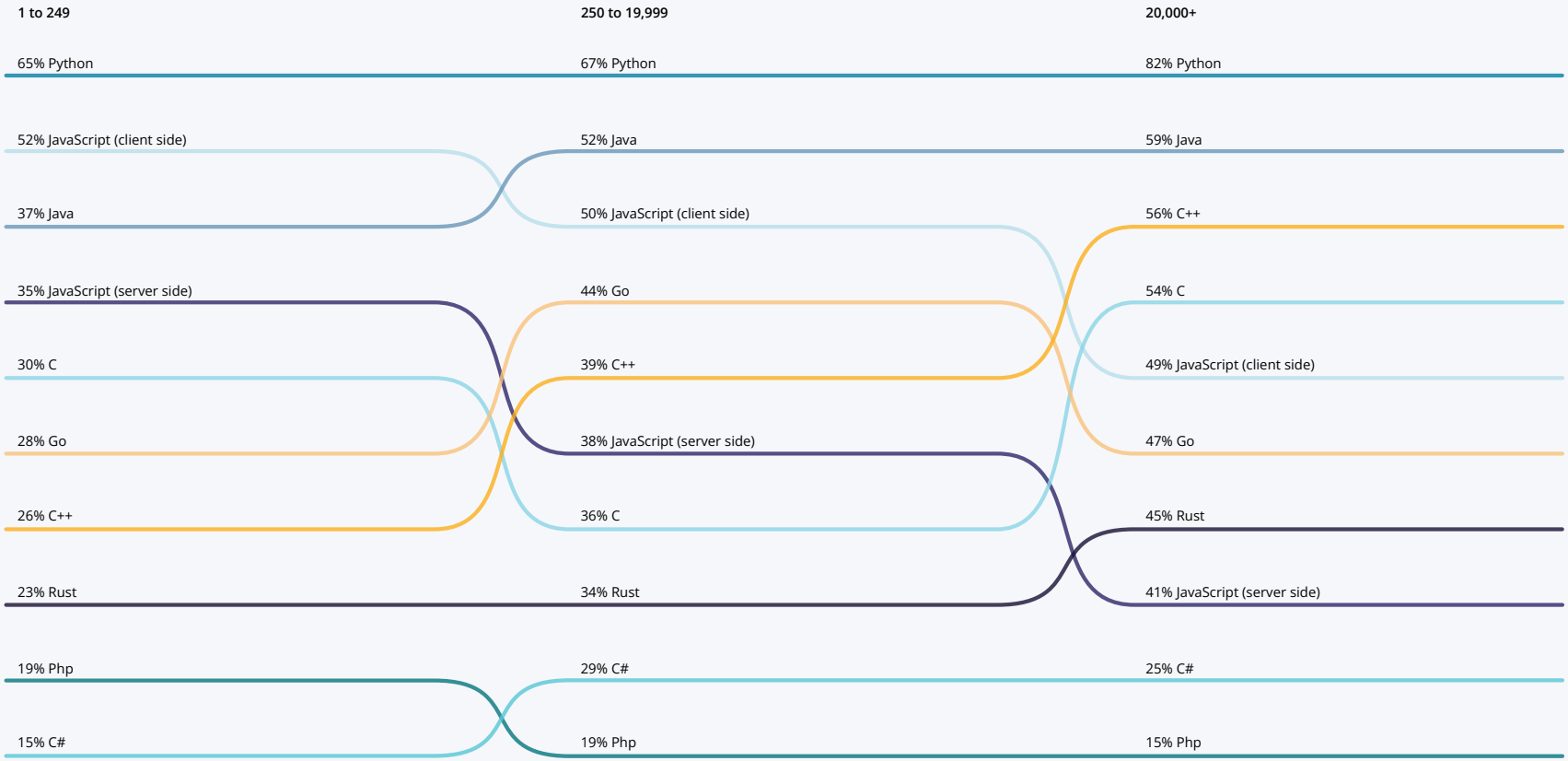


2024 SecEd Survey, Q23 by Q10, Sample Size = 321, Total Mentions = 1,330, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 38

LANGUAGE-SPECIFIC COURSES BY ORGANIZATION SIZE

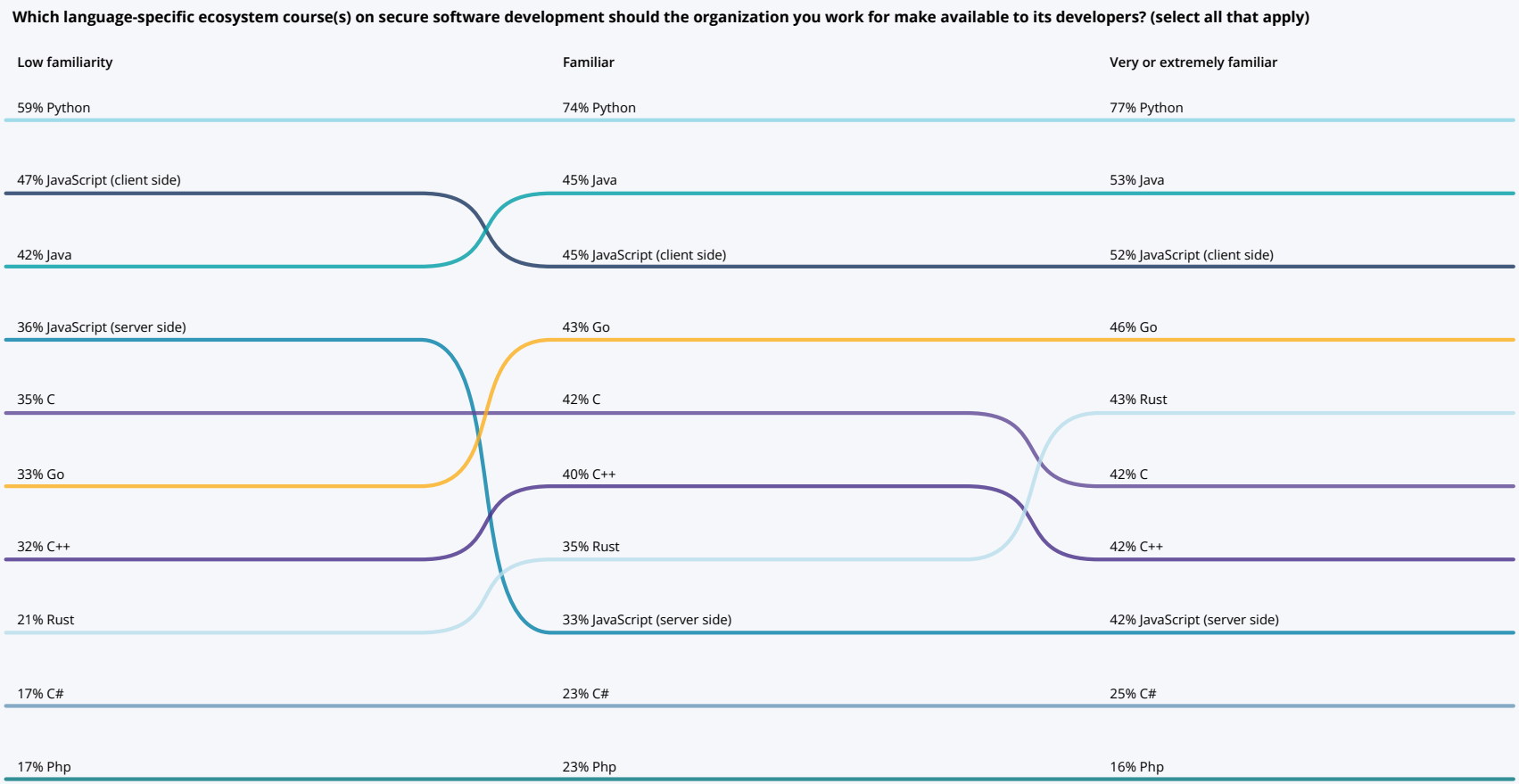
Which language-specific ecosystem course(s) on secure software development should the organization you work for make available to its developers? (select all that apply)



2024 SecEd Survey, Q23 by Q12, Sample Size = 315, Total Mentions = 1,315, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 39

LANGUAGE-SPECIFIC COURSES BY FAMILIARITY

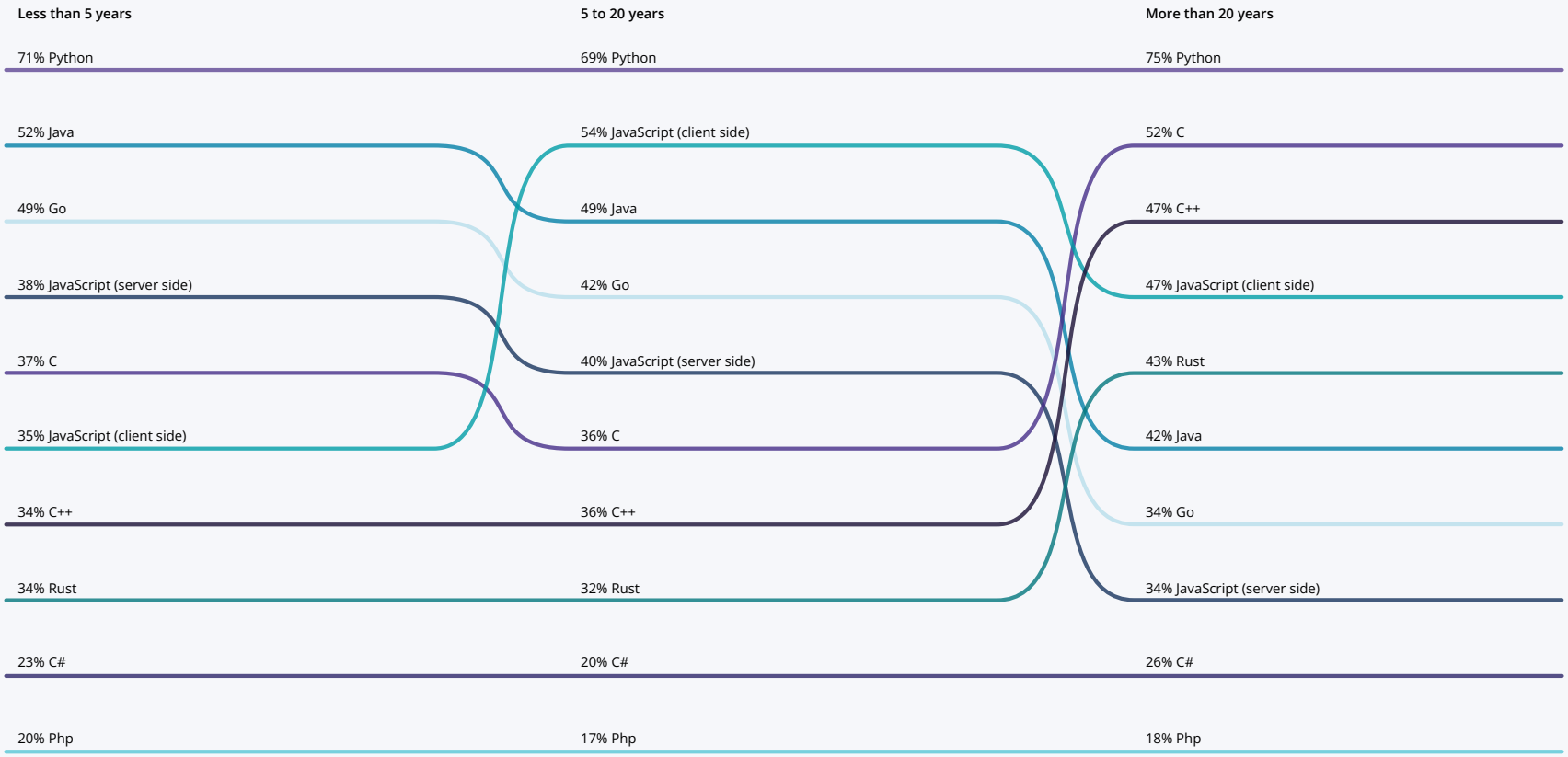


2024 SecEd Survey, Q23 by Q14, Sample Size = 350, Total Mentions = 1,448, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 40

LANGUAGE-SPECIFIC COURSES BY YEARS OF EXPERIENCE

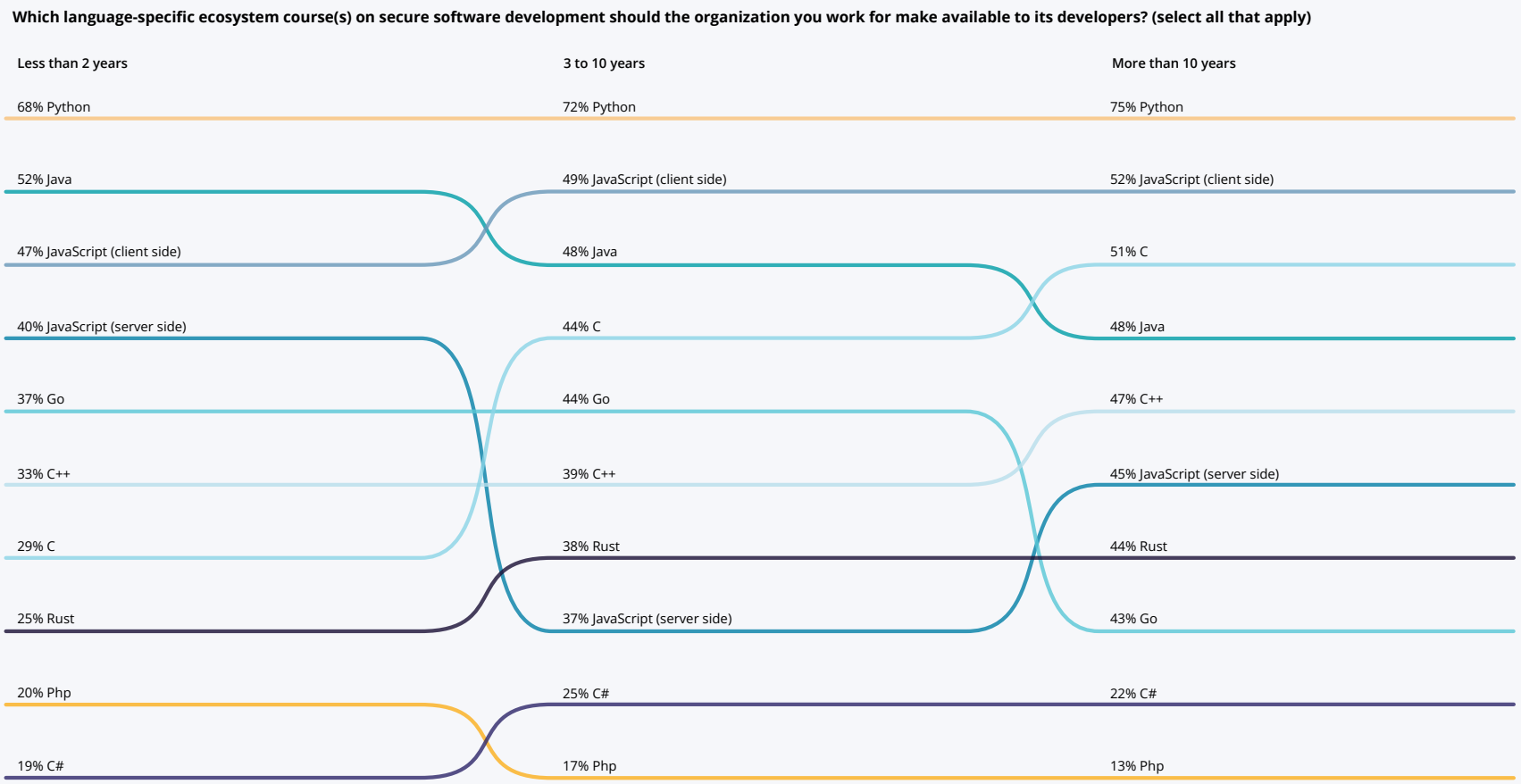
Which language-specific ecosystem course(s) on secure software development should the organization you work for make available to its developers? (select all that apply)



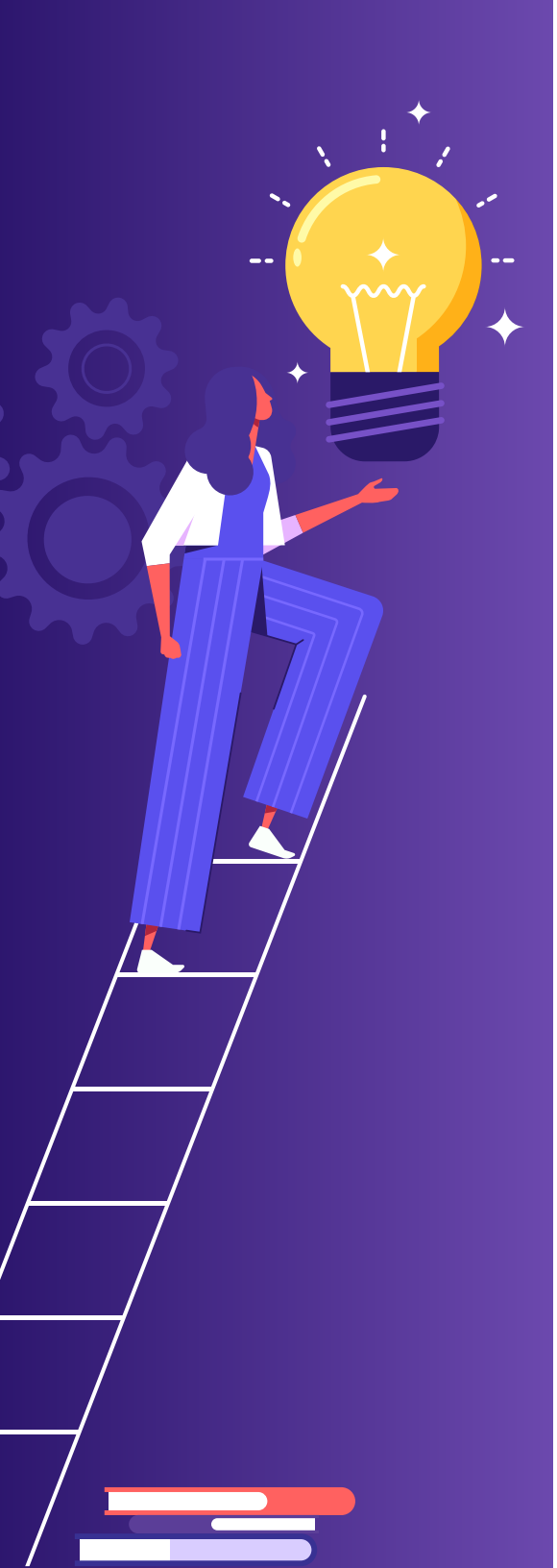
2024 SecEd Survey, Q23 by Q15, Sample Size = 350, Total Mentions = 1,445, the number in front of the name represents the percentage of respondents, each column is sorted by this number

FIGURE 41

LANGUAGE-SPECIFIC COURSES BY YEARS IN SECURE SOFTWARE DEVELOPMENT



2024 SecEd Survey, Q23 by Q15, Sample Size = 328, Total Mentions = 1,377, the number in front of the name represents the percentage of respondents, each column is sorted by this number

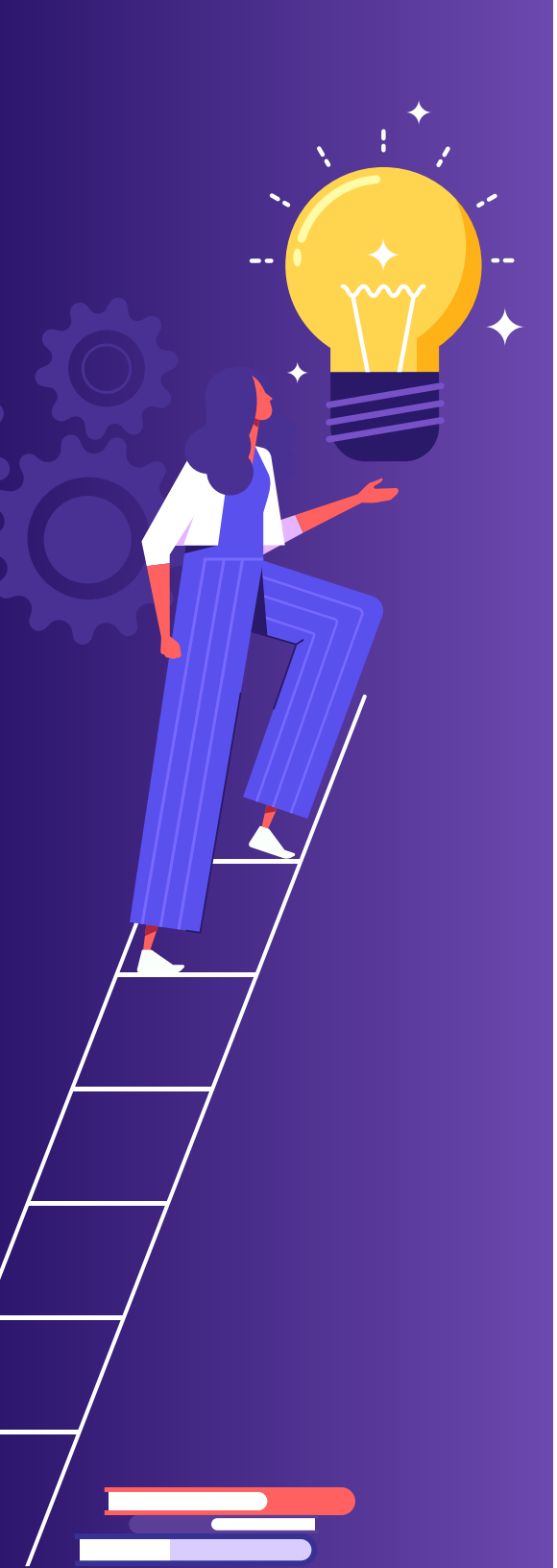


About the Authors

Dr. MARCO A. GEROSA is a full professor of Computer Science at Northern Arizona University and a research analyst at LF Research. His research on software engineering and open source software has resulted in over 200 publications in top-tier venues. He serves on the program committee of renowned conferences and as a reviewer for several journals. Dr. Gerosa has a Ph.D., a master's in Informatics, and a B.S. in Computer Engineering. He is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE) and the Association for Computing Machinery (ACM). He supervised several Ph.D. and M.Sc. students who are now researchers in top institutions. He also has more than 20 years of teaching experience. For more information, visit **http://www.marcoagerosa.com**.

Dr. DAVID A. WHEELER is an expert on OSS and developing secure software. His work on developing secure software includes Secure Programming HOWTO, the OpenSSF Secure Software Development Fundamentals Courses, and Fully Countering Trusting Trust through Diverse Double-Compiling. He is the Director of Open Source Supply Chain Security at the Linux Foundation and teaches a graduate course in developing secure software at George Mason University. Dr. Wheeler has a Ph.D. in Information Technology, a master's in Computer Science, a certificate in Information Security, a certificate in Software Engineering, and a B.S. in Electronics Engineering, all from George Mason University. He is a Certified Information Systems Security Professional and a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE). He lives in Northern Virginia.

STEPHEN HENDRICK is vice president of research at the Linux Foundation, where he is the principal investigator on a variety of research projects core to the Linux Foundation's understanding of how OSS is an engine of innovation for producers and consumers of IT. Steve specializes in primary research techniques developed over 30 years as a software industry analyst. Steve is a subject-matter expert in application development and deployment topics, including DevOps, application management, and decision analytics. Steve brings experience in a variety of quantitative and qualitative research techniques that enable deep insight into market dynamics and has pioneered research across many application development and deployment domains. Steve has authored over 1,000 publications and provided market guidance through syndicated research and custom consulting to the world's leading software vendors and high-profile start-ups.



Acknowledgments

We thank all of the people who participated in the survey process as well as those who dedicate time and effort toward improving secure software development education. Special thanks to Look Left Marketing and Linux Foundation colleagues for their involvement in the various stages of the research process, including:

- Omkhar Arasaratnam
- Jennifer Bly
- Sally Cooper (Look Left Marketing)
- Anna Hermansen
- Hilary Carter
- Adrienn Lawson
- Noah Lehman
- Angelah Liu
- Geena Pickering (Look Left Marketing)
- Chris Poisson (Look Left Marketing)
- Bryan Scanlon (Look Left Marketing)
- David Sprague (Look Left Marketing)
- Jennifer Tanner (Look Left Marketing)
- Harry Toor

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

The **Open Source Security Foundation (OpenSSF)** is a cross-industry initiative by the Linux Foundation that brings together the industry's most important open source security initiatives and the individuals and companies that support them. The OpenSSF is committed to collaborating and working upstream and with existing communities to advance open source security. For more information, please visit us at openssf.org.

THE LINUX FOUNDATION | Research

Founded in 2021, **Linux Foundation Research** explores the growing scale of open source collaboration, providing insight into emerging technology trends, best practices, and the global impact of open source projects. Through leveraging project databases and networks, and a commitment to best practices in quantitative and qualitative methodologies, Linux Foundation Research is creating the go-to library for open source insights for the benefit of organizations the world over.

x.com/linuxfoundation

facebook.com/TheLinuxFoundation

linkedin.com/company/the-linux-foundation

youtube.com/user/TheLinuxFoundation

github.com/LF-Engineering

To reference this work, please cite as follows: Marco Gerosa, David A. Wheeler, Stephen Hendrick, "Secure Software Development Education Study: Understanding Current Needs," foreword by Christopher Robinson and Dave Russo, The Linux Foundation, June 2024.