Zend
by Perforce

2024 PHP Landscape Report

Table of Contents

Foreword

Matthew Weier O'Phinney
Senior Product Manager
Zend by Perforce

To the reader,

The PHP 8 lifecycle is well under way, with conversations already starting about whether there will be another PHP 8 version following 8.4, or if the project will pivot to a new major version. Being mid-way through the PHP 8 lifecycle, it's a good time to see how adoption trends versus previous major versions.

When PHP 7 released, a large part of the ecosystem stayed behind on PHP 5; however, with PHP 8, we're seeing much faster adoption: instead of a close to 2:1 split between EOL and supported PHP as reported last year, the split is much closer to 1:1, with usage of PHP versions prior to 7.4 dropping dramatically. The implication is pretty clear: the migration path between PHP 7 and PHP 8 has been much easier to accommodate.

That said, there are other factors informing this trend. Security, and in particular supply chain security, are at the top of many organizations priorities. One way to keep secure is to keep on current software. While businesses can choose long term support runtimes, sometimes this doesn't address the rest of the application: frameworks and libraries may only provide security patches for up-to-date runtime versions. Another factor in PHP 8 adoption is the high adoption of containers and orchestration practices within the PHP ecosystem. Both technologies simplify the process of testing and deploying new runtime versions, allowing companies to migrate more quickly. Additionally, many of the recent changes in the language itself are either security oriented or aid in creation of maintainable software: better randomization (which allows for better data hashing and encryption), data masking (via the `#[SensitiveParameter]` attribute), and read-only properties and classes (which enable immutable data structures).

What I am observing in the ecosystem is a return to a basic principle of PHP: scalability. Containerization and orchestration practices focus on the fact that PHP is horizontally scalable by design, via its "shared nothing" architecture; both allow businesses to add more servers as demand increases, and scale down when it dissipates. We also observe an increase in adoption of technologies that help companies scale existing architecture: message queues, job queues, and more. These technologies do not come without cost; we observe in the report that these are more widely adopted in larger organizations, which can more readily train their teams, or hire experts. But overall, they help companies trim costs at a time when many are feeling the pinch.

The conclusion is positive: PHP remains a fantastic choice for web-facing applications and APIs, due to its wide-ranging integrations, evolving syntax, and focus on security.

Enjoy the report,

Matthew Weier O'Phinney
Senior Product Manager
Zend by Perforce

About the Survey

The Zend 2024 PHP Landscape Report is based on the results of an anonymous survey of self-identified PHP users and administrators that was conducted between the months of October and December in 2023. The survey received a total of 572 valid responses.

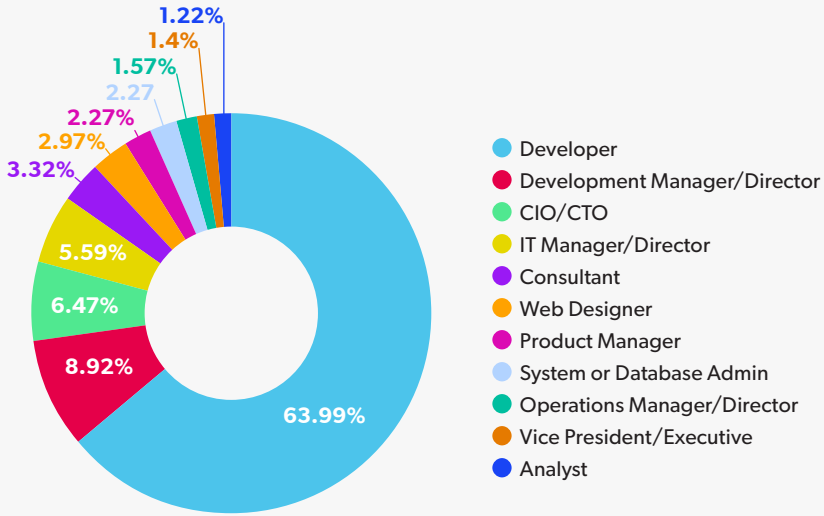
RESPONDENT FIRMOGRAPHICS

In our first question of the survey, we asked respondents to identify their job title. As in previous years, the bulk of respondents were in development focused roles, with nearly 64% developers. Rounding out the top three, Development Manager/Director, and CIO/CTO roles represented 8.92% and 6.47% of respondents, respectively.

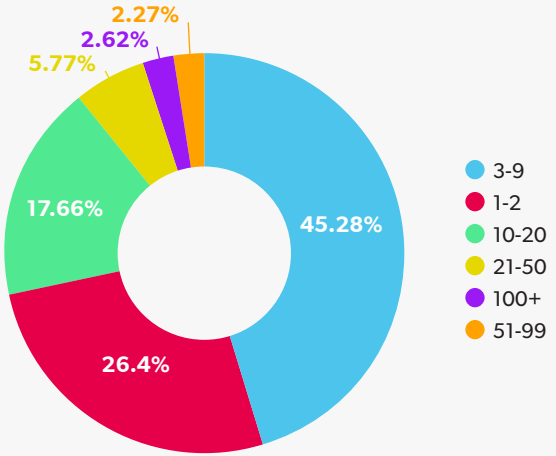
DEVELOPMENT TEAM SIZE

In our next question, we asked respondents to share the size of their development teams. Development team size representation stayed largely the same year over year as well, with teams with under 10 developers representing nearly 74% of respondents, with the bulk of respondents (45.28%) having development teams with between 3-9 developers.

WHICH JOB TITLE BEST MATCHES YOUR CURRENT ROLE?



WHAT IS THE SIZE OF YOUR DEVELOPMENT TEAM?



COMPANY SIZE

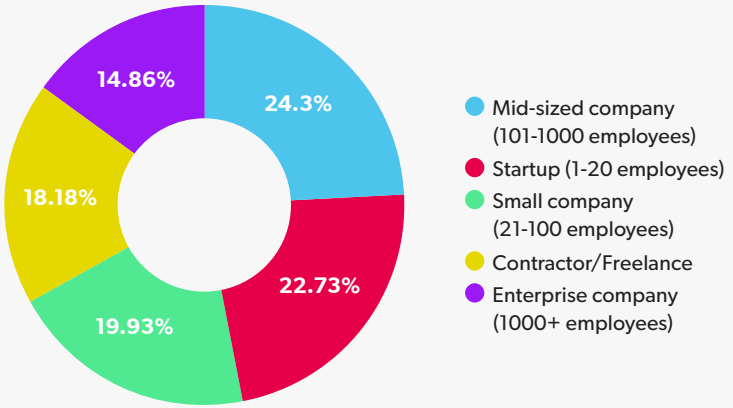
Next we asked teams to share the size of their company. We saw fairly even distribution of company sizes, with 24.3% of respondents working for mid-sized companies, 22.73% working for startups, and 19.93% working for small companies. Contractors / Freelancers and Enterprise Companies made up the remaining respondents, with 18.18% and 14.86% respectively.

REGION

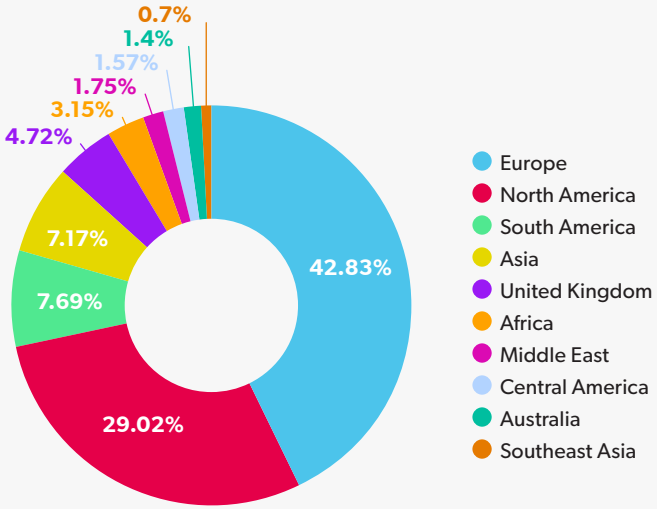
Next, in a new question for this year's survey, we asked teams to identify the location of their business headquarters. Europe (42.83%) and North America (29.02%) had the largest representation in our survey. South America, Asia, and the United Kingdom rounded out the top five with 7.69%, 7.17%, and 4.72% of responses, respectively.

> Europe (42.83%) and North America (29.02%), had the largest representation in our survey.

WHAT IS THE SIZE OF YOUR COMPANY?



IN WHICH GLOBAL REGION ARE YOUR BUSINESS HEADQUARTERS?



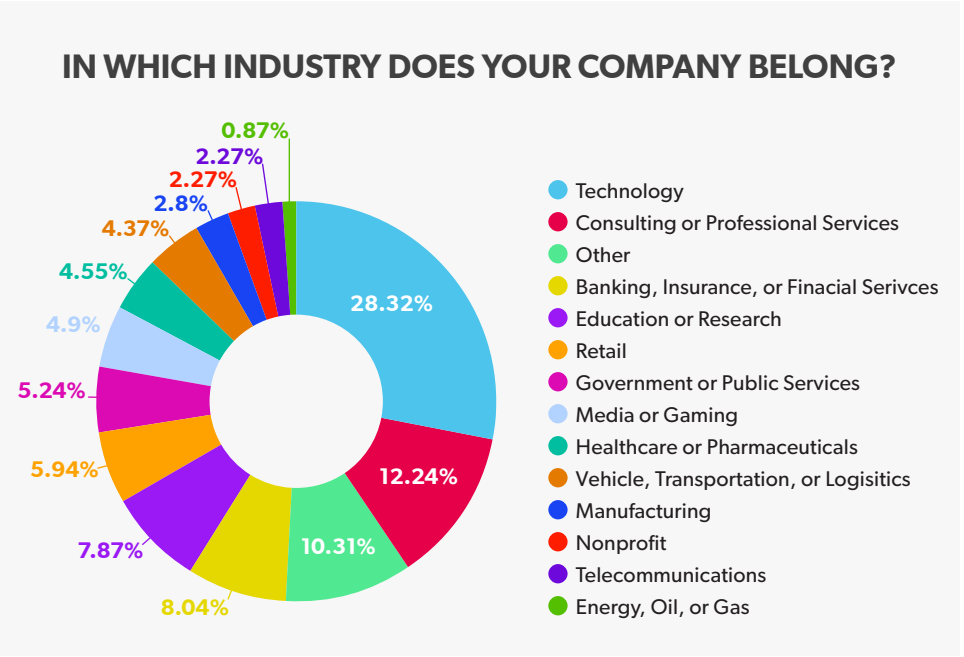
INDUSTRY

In our final firmographic question, we asked respondents to self-select the industry in which their company operates. The top industry selected in this year's survey was Technology at 28.32%, followed by Consulting or Professional Services at 12.24%. Among respondents that selected Other, marketing was the top write-in response.

KEY TAKEAWAYS

Overall, we had a balanced firmographic profile this year. As in previous years, our respondents skewed toward development focused roles, which makes sense considering developers are in roles that are often hands on in developing and managing PHP-based applications.

Given the sample size, and the even distribution of respondents among varying company sizes, we feel that the 2024 PHP Landscape Report is highly representative of teams working with PHP around the world and gives accurate insight into the state of PHP development and management.



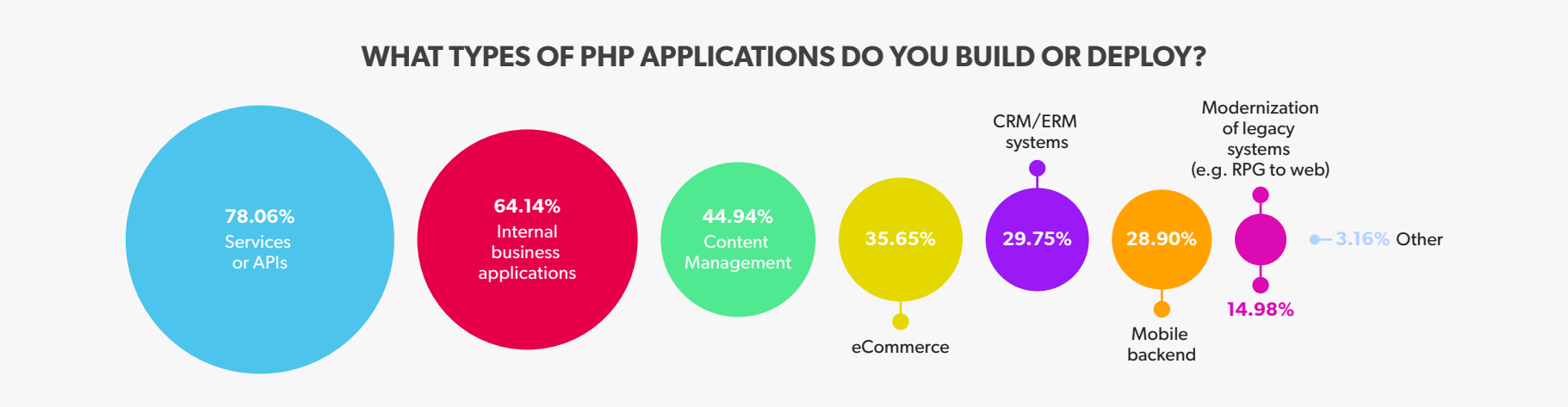
PHP APPLICATION DEVELOPMENT TRENDS

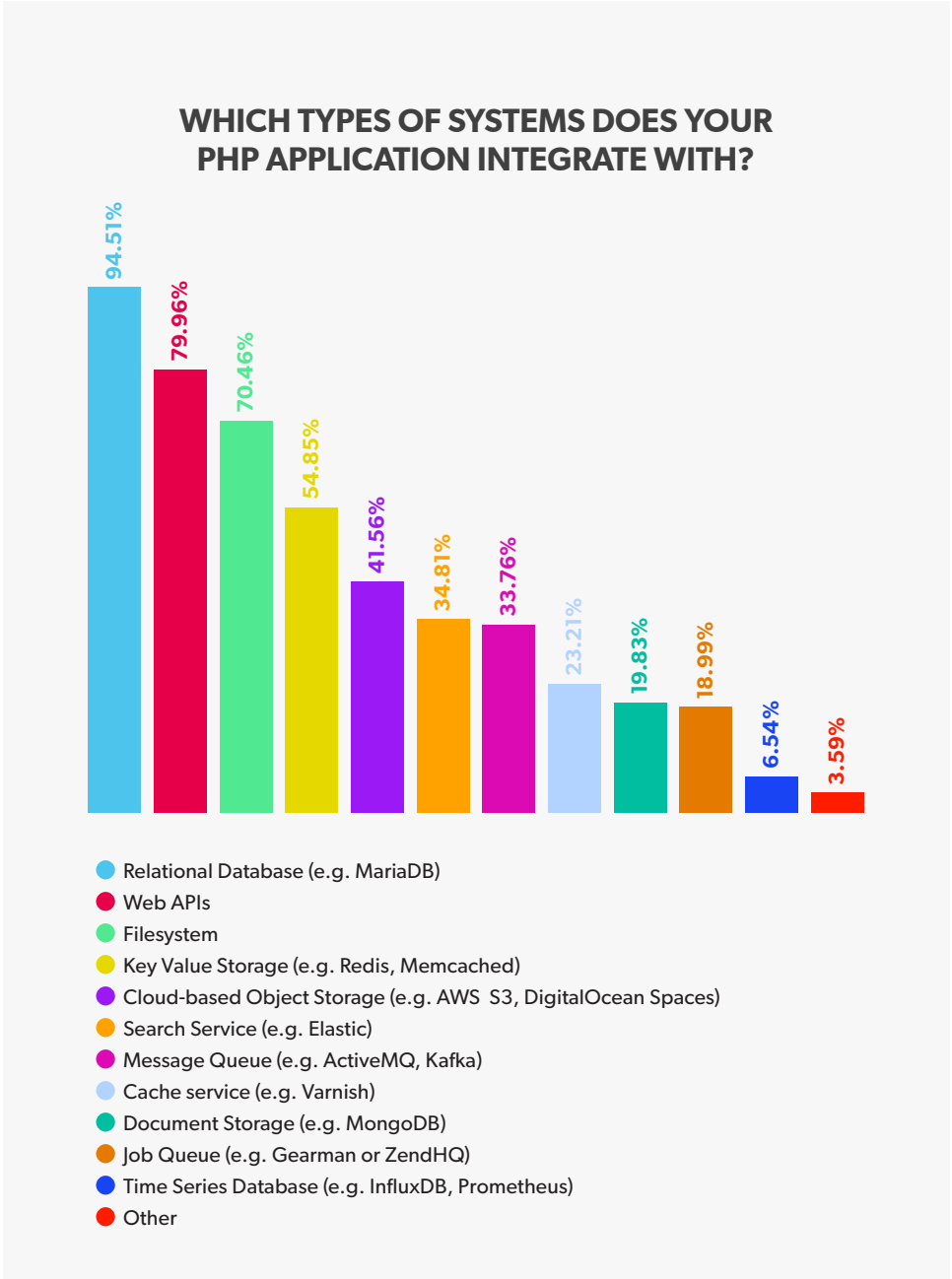
In our next set of questions, we asked respondents to share more information about the types of PHP applications they develop, the types of technologies that they integrate with and use, and where they deploy their application. We also asked a handful of questions regarding container and container orchestration technology adoption, and the top container and container orchestration technologies used in PHP applications today.

PHP APPLICATION TYPES

In our first question for this section, we asked teams to share the types of PHP applications they build or deploy, with respondents able to select multiple entries as applicable.

As with the previous two years of our survey, the top three application categories were Services or APIs (78.06%), Internal Business Applications (64.14%), and Content Management Systems (44.94%). Rounding out the top five were eCommerce (35.65%), and CRM / ERM Systems (29.75%).





PHP APPLICATION INTEGRATION TRENDS

Next, we asked respondents to identify the types of systems their PHP applications integrate with. Unsurprisingly, 94.51% of respondents noted that their PHP applications integrate with Relational Databases. Web APIs were the next most common selection with 79.96% of respondents, followed by Filesystems in third at 70.46%.

In our 2023 report, we noted that the breadth of technologies that PHP developers now integrate within their applications is increasing, and we noticed that trend continue in 2024. Year over year, we saw growth in the percentage of respondents integrating their applications with nearly every category, including Relational Databases, Web APIs, Filesystems, Key-Value Storage, Search Services, and Cache Services. A new entry this year, Job Queue, noted 18.99% integration among respondents.

KEY TAKEAWAYS

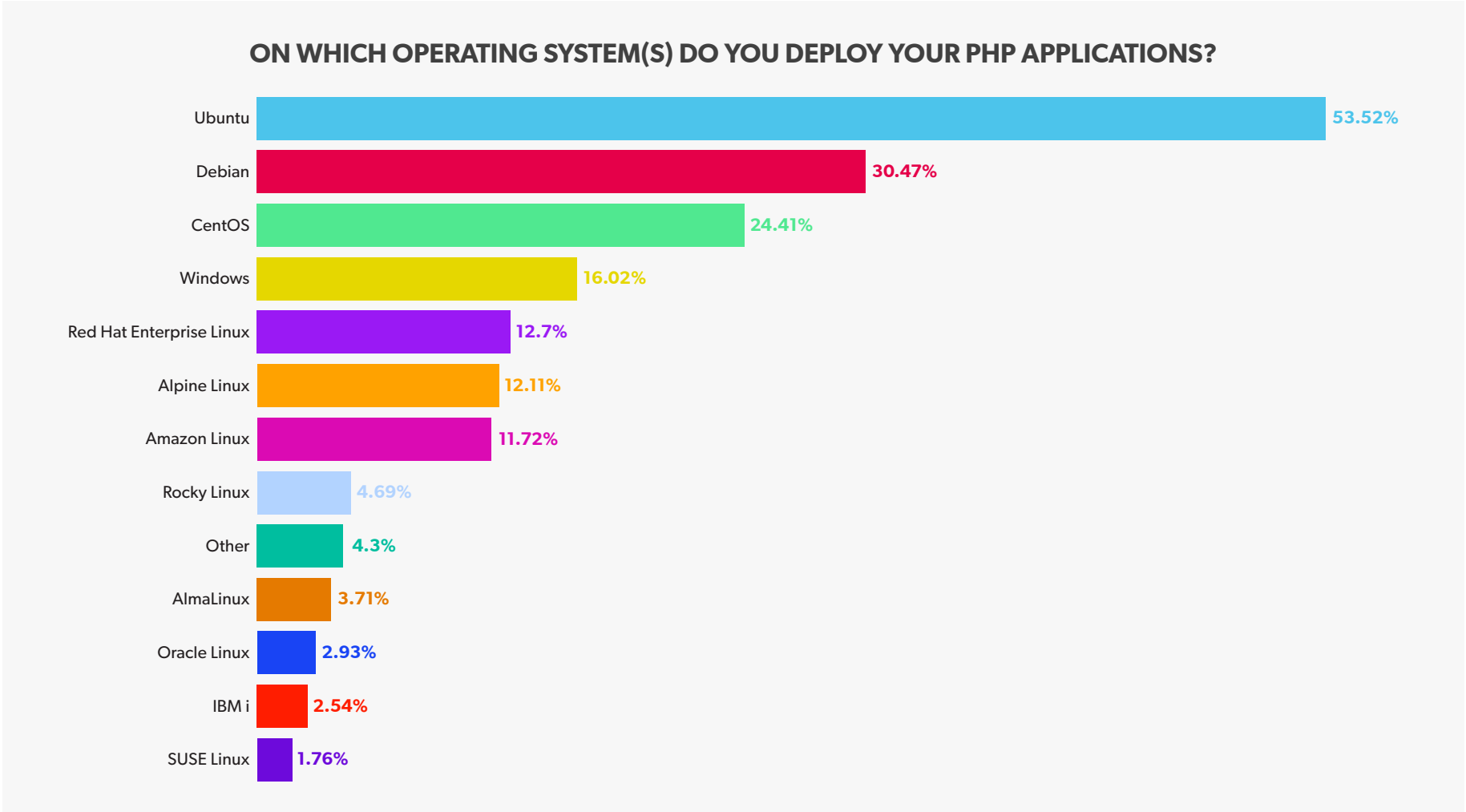
PHP is a glue language for the web, and its wealth of integration extensions give huge flexibility for organizations and developers. When looking through the options, one thing to note is that almost all of these provide either some form of data storage or process deferment; combined, these factors enable scalability in web applications.

This is the other factor in PHP's ongoing popularity and suitability for the web: horizontal scaling. You can spin up more application instances when you need to scale your application up, instead of taking down the application to provision a larger, more capable server. Horizontal scalability helps reduce total cost of ownership, particularly when deploying to the cloud, as you can scale back when your traffic goes down, reducing your costs.

OPERATING SYSTEM USAGE TRENDS

Next, we asked respondents to share the operating systems they use in deploying their PHP applications. In an effort to improve accuracy among teams deploying on multiple operating systems, we allowed respondents to select multiple operating systems as applicable.

Our 2024 survey found Ubuntu to be the most-used operating system at 53.52%, followed by Debian at 30.47% and CentOS at 24.41%. Year over year, CentOS usage dropped in ranking from the second most-used distribution in 2023 to the third most-used distribution in 2024, signaling teams departing the once-dominant, and soon-to-be discontinued open source distribution for comparable paid, or open source distributions like RHEL, Rocky Linux, or AlmaLinux.

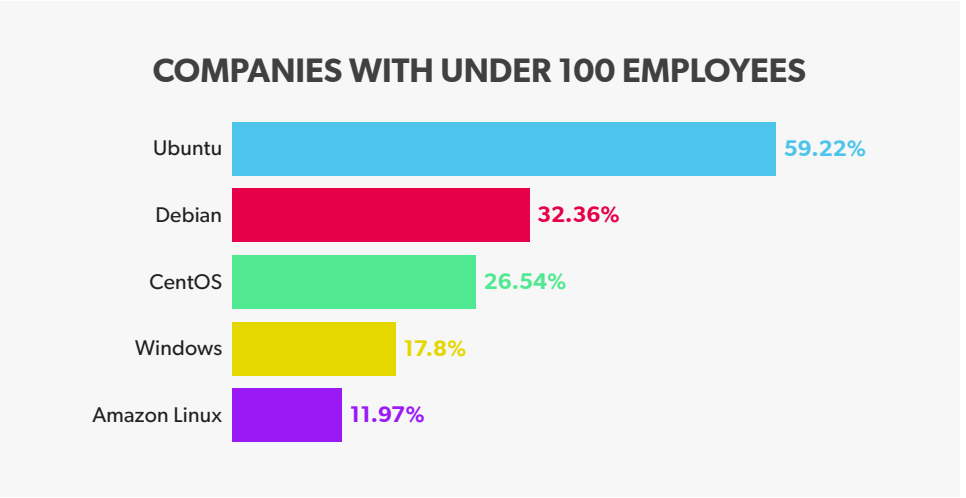


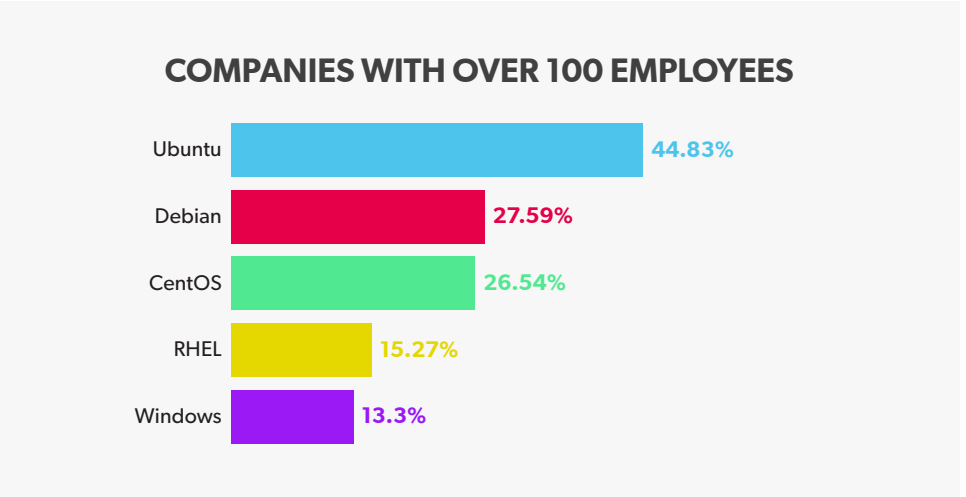
When looking at the most-used operating systems by company size, we found that larger companies (15.27%) adopted RHEL more often than smaller companies (11%).

KEY TAKEAWAYS

A decade ago, RHEL and CentOS were the hands-down front runners for production web application deployment. Today, we're seeing the Debian ecosystem in the lead by a large margin. One cannot help but wonder if the moves by RedHat in the past few years to halt development of 1:1 compatibility of CentOS with RHEL releases have led to distrust of that ecosystem by consumers. One troubling takeaway from the stats is that CentOS is still in 3rd position, considering that its last supported version, version 7, reaches end of life June 30, 2024. Considering the relatively small install bases of RHEL and other RHEL-compatible distributions, we are curious to see if organizations will migrate within the RHEL ecosystem, or to one of the Debian-based distributions this year.

Another item to note is that Alpine Linux is almost at parity with RHEL. Alpine is rarely used as a base operating system other than in containers, and its rise in usage in the PHP ecosystem is very likely directly related to the popularity of containers for PHP application deployment.

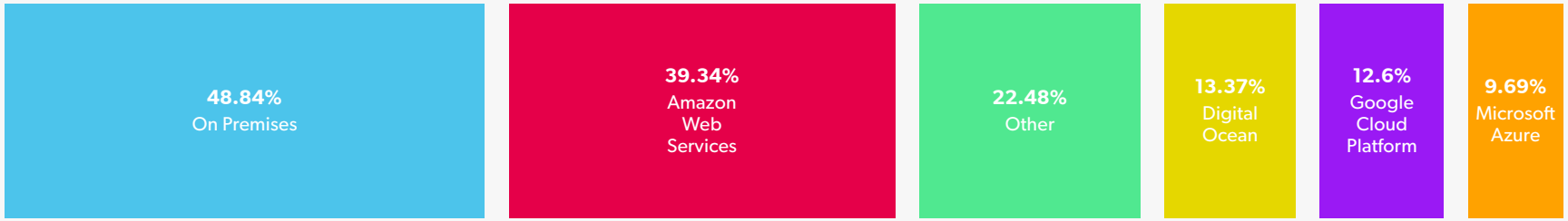




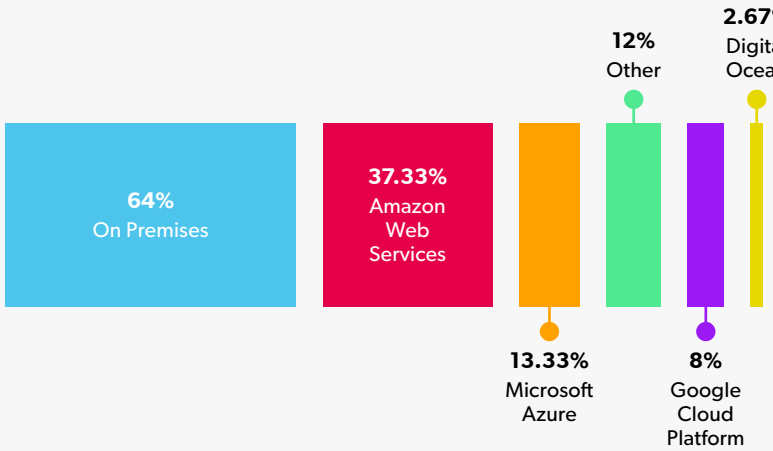
DEPLOYMENT TRENDS

In our next question, we asked respondents to share where they deploy their PHP applications, with the option to select multiple options as applicable. At the top of the list for 2024 was On-Premises, with 48.84% of respondents noting that they have at least one on-premises deployment. Running down the list, Amazon Web Services and "Other" represented 39.34% and 22.48% of respondents, respectively. Among those who selected "Other," respondents listed a number of popular hosting providers, and less common cloud platforms.

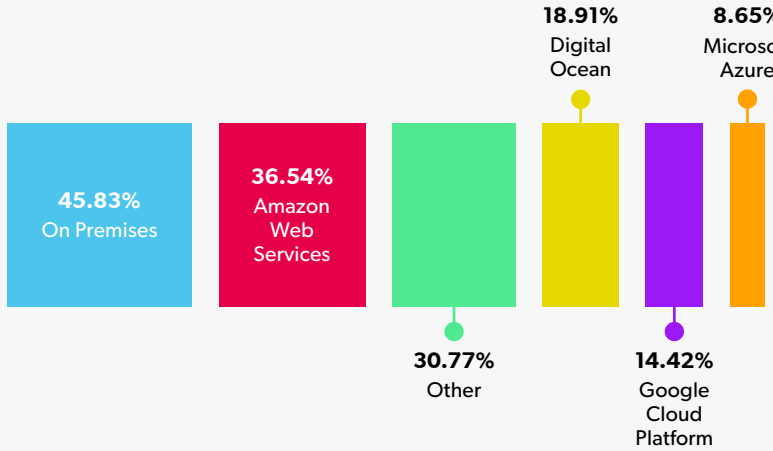
WHERE ARE YOU DEPLOYING YOUR PHP APPLICATIONS?



COMPANIES WITH OVER 100 EMPLOYEES

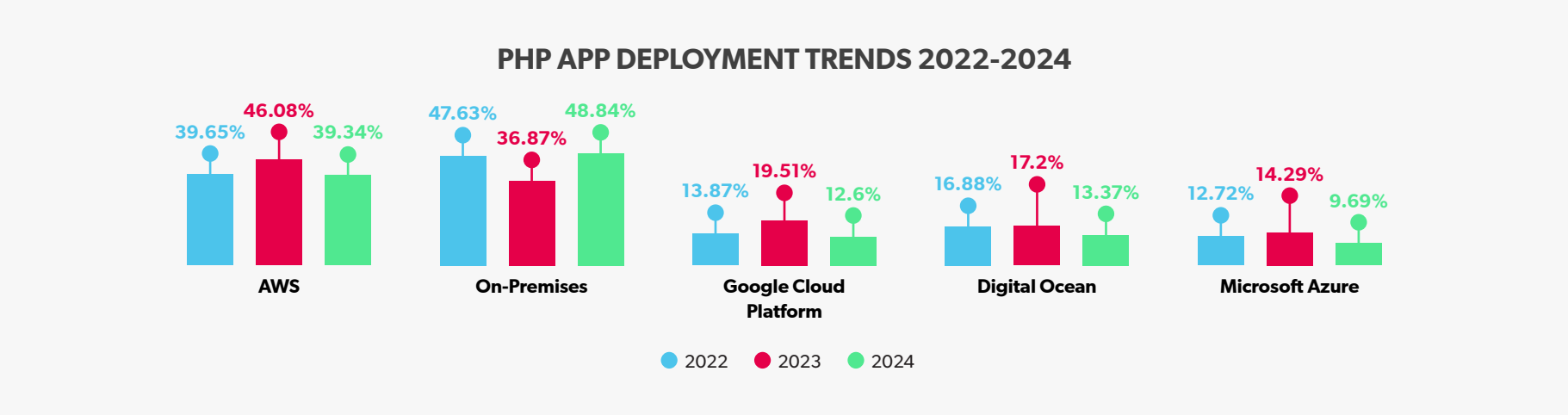


COMPANIES WITH UNDER 100 EMPLOYEES



When looking at responses by company size, we noted that large companies were more likely than small companies to deploy on premises (64% vs. 45.83%).

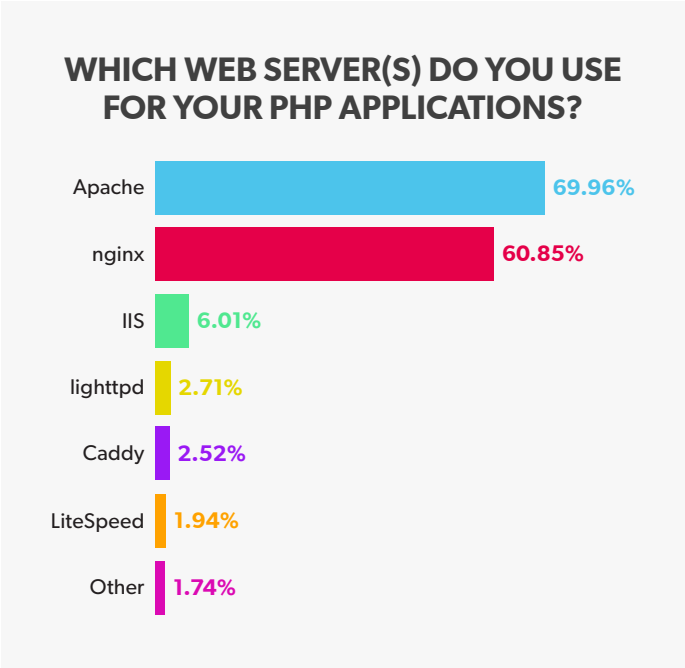
Looking at deployment choices dating back to our 2022 survey, we noted that on-premises deployments reached their highest rate in the last three years, while AWS, Google Cloud Platform, Digital Ocean, and Microsoft Azure dropped to their lowest usage rates of the last three years.



KEY TAKEAWAYS

In the 2023 PHP Landscape Report, we saw a cloud provider, AWS, take the top spot for PHP application deployment for the first time; this year has reversed that trend — and in a quite significant way, as this is the highest number of respondents in the past three years indicating on premise deployment as their choice. In talking with customers and prospects, we hear many discuss moving back to on premise deployment as a way to reduce operations costs, as well as secure business and customer data. This choice is clearly more popular with larger companies, where they have the space and personnel to run these data centers.

For those companies choosing to deploy to the cloud, there's a clear bifurcation of preference based on organization size. While AWS is the lead choice regardless of company size, smaller companies show a preference for Digital Ocean over more "enterprise-y" clouds such as Azure and GCP.



WEB SERVER TRENDS

In our next question, we asked respondents to share which web servers they use for their PHP applications, with the option to select all that apply. The top choice among application servers in our survey was Apache, at 69.96% of responses, followed by nginx at 60.85%. IIS, which simultaneously jumped in ranking to number three and decreased in percentage use year over year, came in at 6.01%.

Year over year, we saw usage of Apache and nginx climb, while use of IIS, lighttpd, LiteSpeed, and Caddy all declined.

KEY TAKEAWAYS

This marks the second year in a row where we have observed relative parity between Apache HTTPD and nginx. What's interesting about this year, however, is that both solutions gained adoption, fully at the expense of other web server solutions.

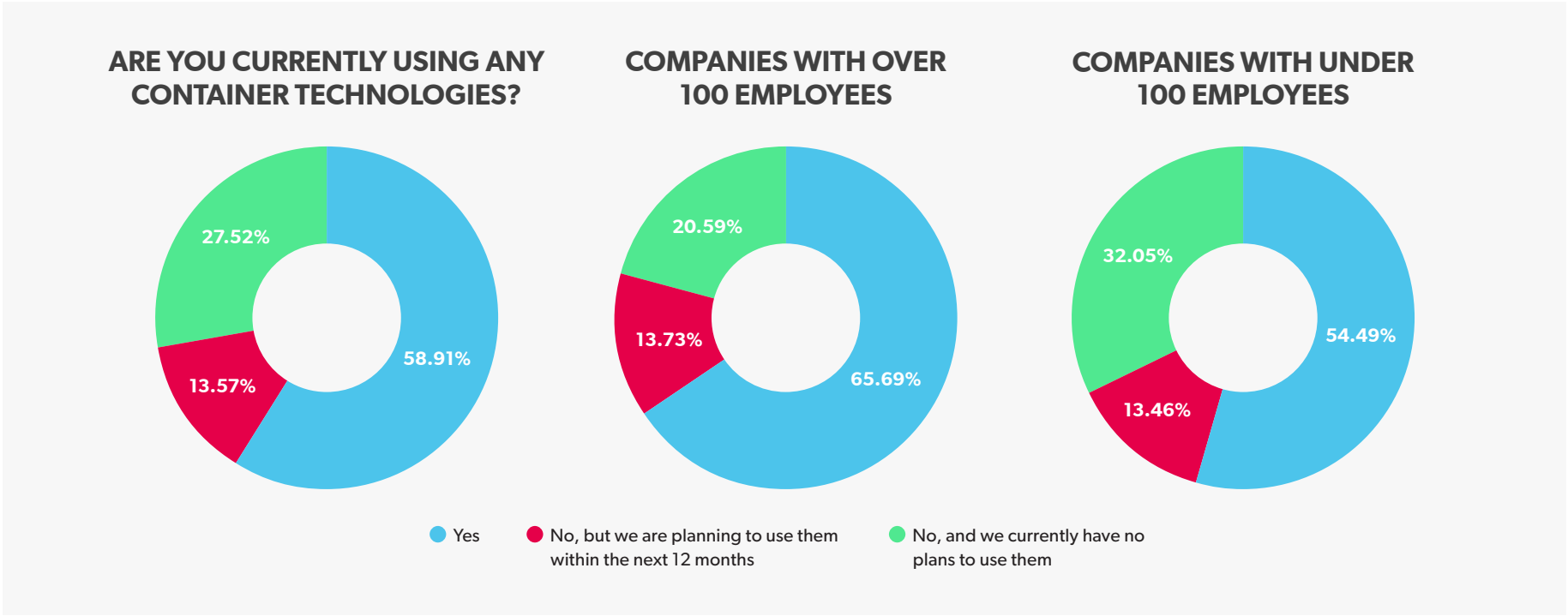
Considering that the FrankenPHP project, which provides a PHP application server on top of Caddy, has reached a stable release in recent months, we are curious if we will see this trend reverse in the coming year, or if these use cases will remain niche.

CONTAINER TRENDS

Next, we turned our attention to the container ecosystem. We asked respondents to share their status and plans around containerization technologies. Our survey found 58.91% of respondents noting that they are currently using container technologies, with an additional 13.57% planning on using container technologies within the next 12 months. 27.52% of respondents reported no plans to use container technologies in the future.

Looking at these results year over year, container technology adoption stayed relatively flat, with a slight increase (57.52% to 58.91%) in those indicating usage of container technologies. Year over year, those planning to use container technologies in the next 12 months fell significantly (19.79% to 13.57%), while those reporting not using container technologies and not having plans to do so increased (22.70% to 27.52%).

As with our 2023 previous survey, we found large companies more likely to indicate usage of container technologies than smaller companies, with 65.69% adoption vs. 54.49% respective adoption.



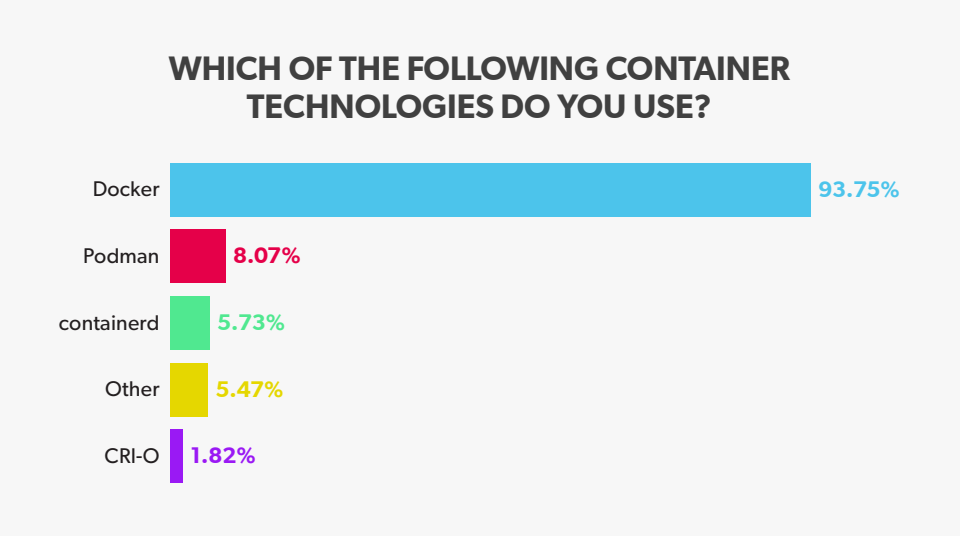
TOP CONTAINER TECHNOLOGIES

Next, we asked those who indicated using container technologies to share which technologies they deploy, with the option to select multiple options as applicable. Our survey found Docker as the top selection at 93.75%, followed by Podman (8.07%), containerd (5.37%), Other (5.47%), and CRI-O (1.82%). Among those who selected Other, the most common write-in responses were FreeBSD, OpenShift, and LXC.

Looking at those choices year over year, we saw a significant uptick in Docker adoption (68.34% to 93.75%), and moderate dips in Podman (10.22% to 8.07%), containerd (9.38% to 5.37%), and CRI-O (8.54% to 1.82%).

KEY TAKEAWAYS

While a clear majority of respondents are using containers or plan to use them, we note that this adoption is higher in larger organizations. When we consider that larger companies are (a) deploying more often on premise, and (b) often have hybrid deployment strategies, containers are a more natural fit and solution to these needs. Additionally, larger organizations are more likely to attract and retain engineers with expertise in container operations.

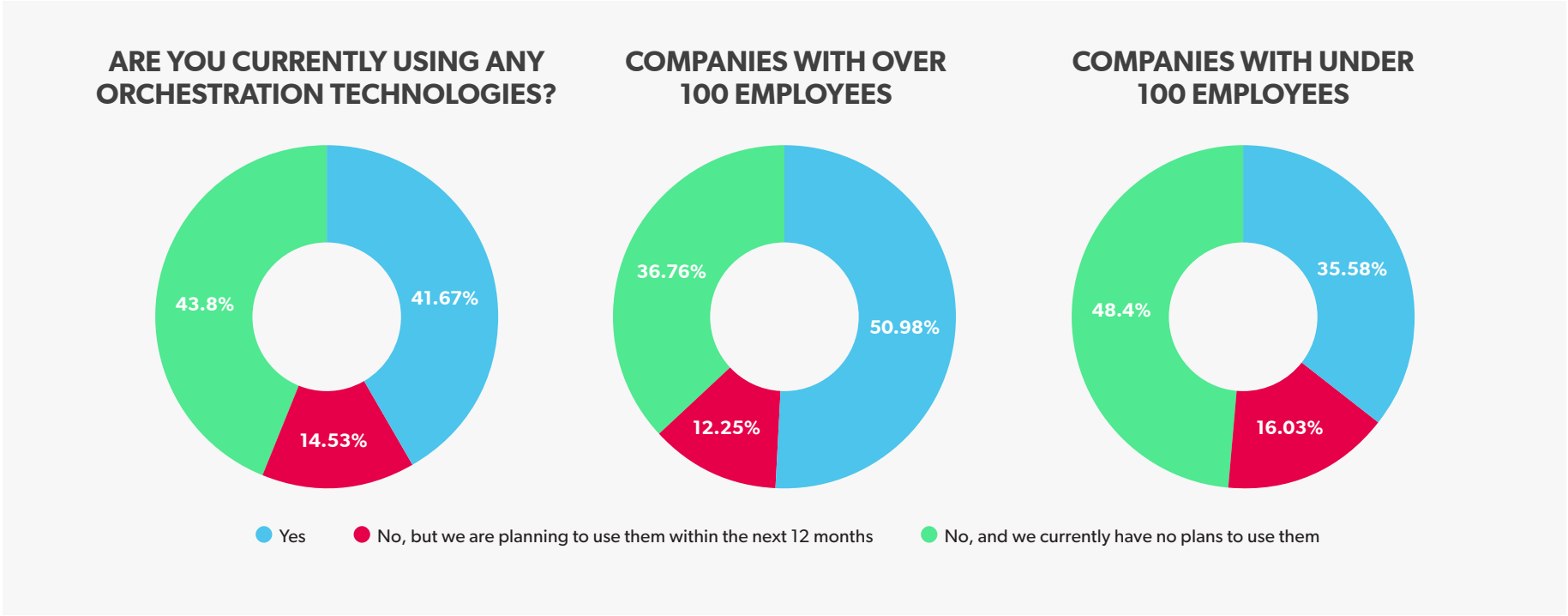


ORCHESTRATION TRENDS

In our next question, we asked respondents to share their status and plans regarding adoption of orchestration technologies. 43.80% of respondents noted that they don't use orchestration technologies and have no plans to do so, while 41.67% indicated that they are currently using them. 14.53% noted that they don't currently use orchestration technologies, but have plans to do so in the next 12 months.

Looking at these results year over year, we noticed a significant shift in responses. Those indicating no, with no plans to use orchestration technologies jumped significantly (35.12% to 43.80%), while those stating current usage fell (47.70% to 41.67%).

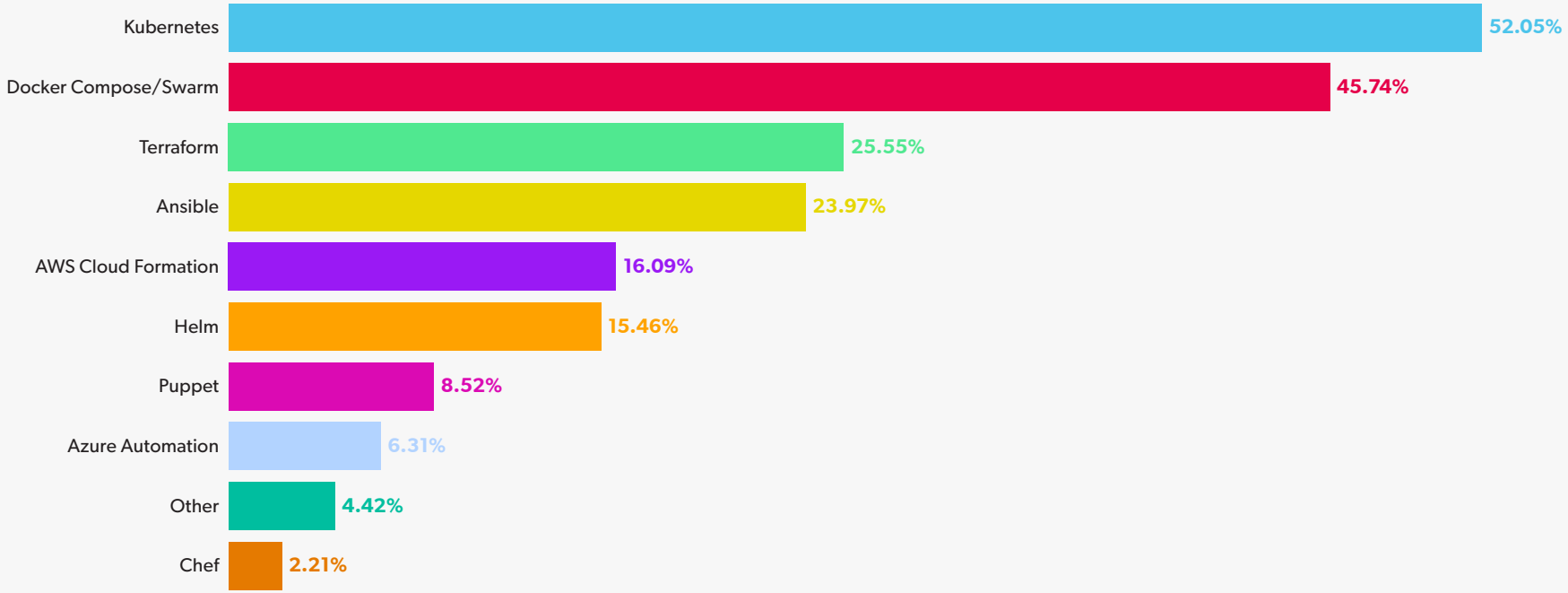
Looking closer, our results indicated that large companies were far more likely to use orchestration technologies, with 50.98% of large companies indicating current usage, compared to only 35.58% of small companies.



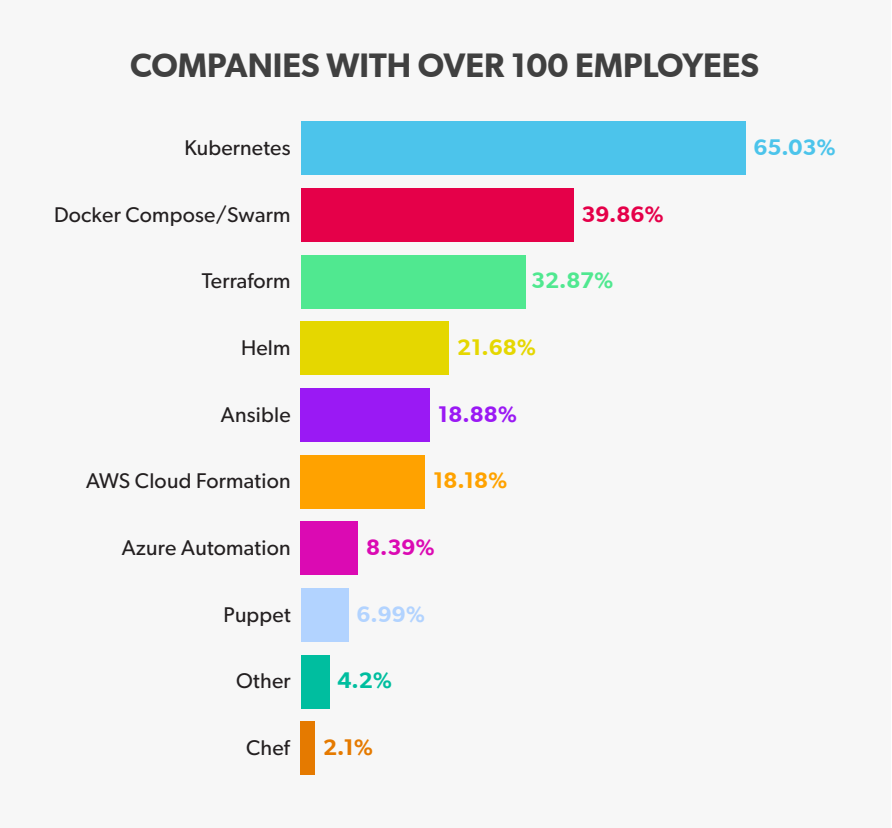
TOP ORCHESTRATION TECHNOLOGIES

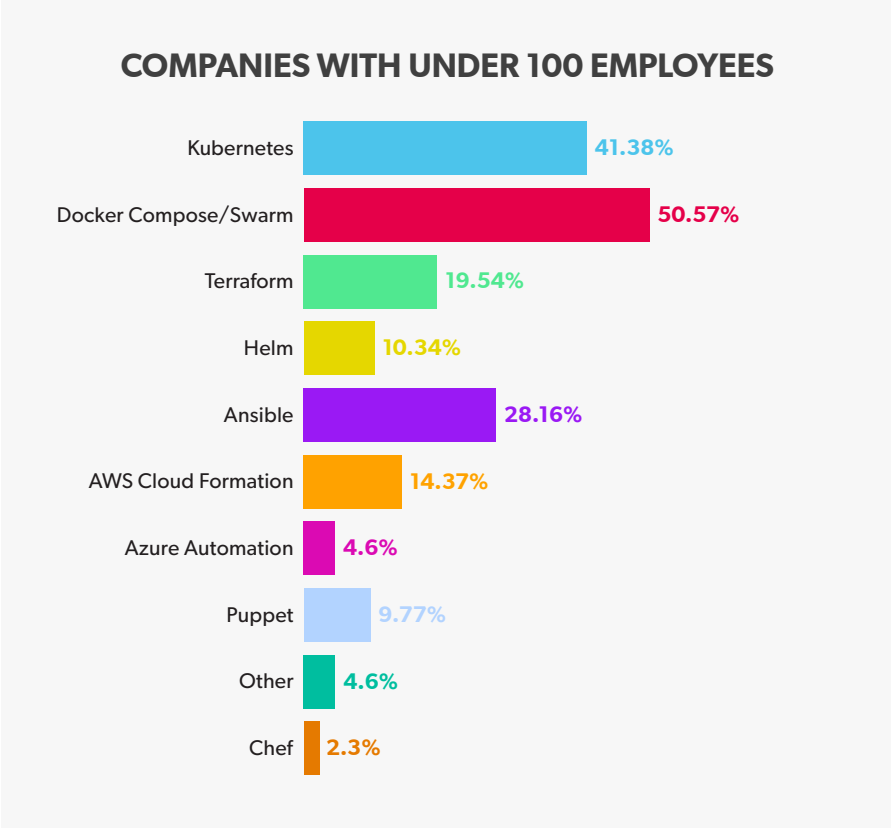
For those that noted use of orchestration technologies, we asked them to share which technologies they use. Top choices included Kubernetes (52.05%), Docker Compose / Swarm (45.74%), and Terraform (25.55%). Rounding out the top five were Ansible and AWS Cloud Formation at 23.97% and 16.09% respectively.

Looking at selections year over year, Kubernetes usage climbed (44.81% to 52.05%), while Ansible (31.90% to 23.97%), Terraform (27.85% to 25.55%), AWS Cloud Formation (25.32% to 16.09%), and Helm (19.49% to 15.46%) all dropped in popularity.



Looking at orchestration tool selection by company size, there were a few interesting shifts. Docker Compose and Swarm was a more popular selection for members of small companies than those in large companies (50.57% vs. 39.86%).





KEY TAKEAWAYS

Our takeaway observation on orchestration trends mirrors our takeaway for containers. With larger organizations and more complex deployment needs, companies need to embrace orchestration technologies. Consensus lies on containers as being the most reliable way to deploy PHP applications, and for this reason we see Kubernetes and Docker Compose / Swarm as the leaders for orchestration technologies in the PHP ecosystem. We find the rise in popularity of Compose / Swarm particularly interesting, as it demonstrates that the simpler paradigm these tools present is a good fit for many PHP applications — particularly in smaller organizations where scaling needs are less.

We also observe that adoption of cloud-specific tooling such as AWS Cloud Formation and Azure Automation has decreased. We feel this is a good trend, as it prevents vendor lock-in, and allows for hybrid cloud deployment opportunities.

SECURITY AND COMPLIANCE TRENDS

In our 2024 survey, we revised and expanded our questions related to security and compliance. With security and compliance a top concern across nearly all development ecosystems, we reformulated our questions to better understand how PHP teams feel about the overall security of their applications, and the measures they take to improve and maintain application security and compliance.

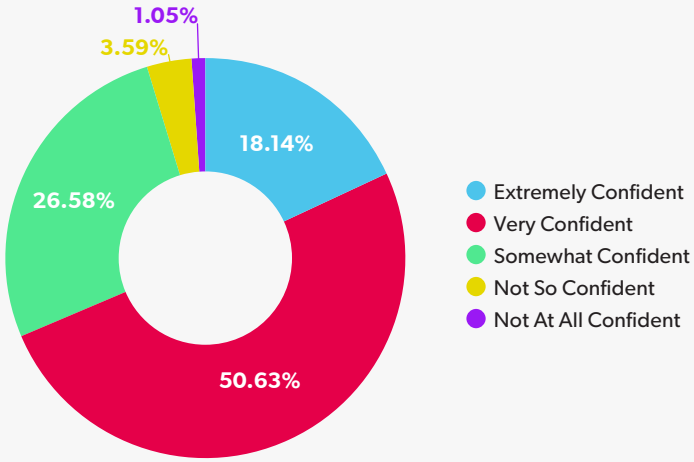
SECURITY TRENDS

For our first addition to this section, we asked respondents to share their confidence level in the security of their PHP applications. As you'll see in the data below, that level of confidence is correlated with other data points from the survey – including respondent job title categories, and the need to meet compliance standards.

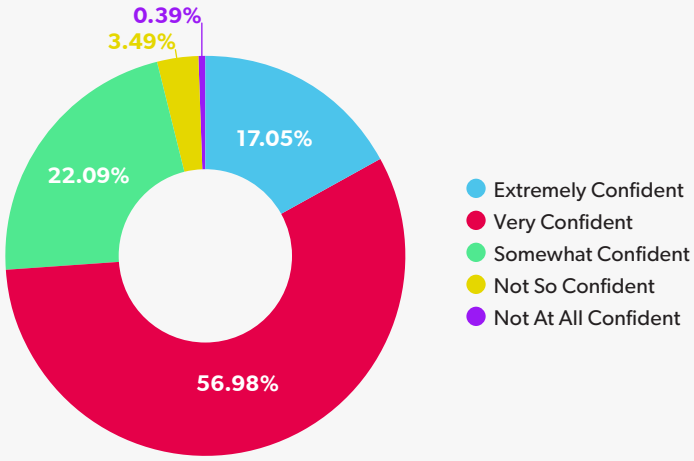
Looking at the unsegmented results, over half of respondents noted they were "Very Confident" in the security of their PHP applications, followed by 26.58% at "Somewhat Confident," and "18.14% "Extremely Confident." Combined, 95.35% of respondents noted positive sentiment around their PHP application security.

When looking at the results for respondents who identified needing to meet compliance requirements, we saw a jump in the percentage (50.63 to 56.98%) of results who were "Very Confident" in their PHP application security.

HOW CONFIDENT ARE YOU IN THE SECURITY OF YOUR PHP APPLICATION(S)?

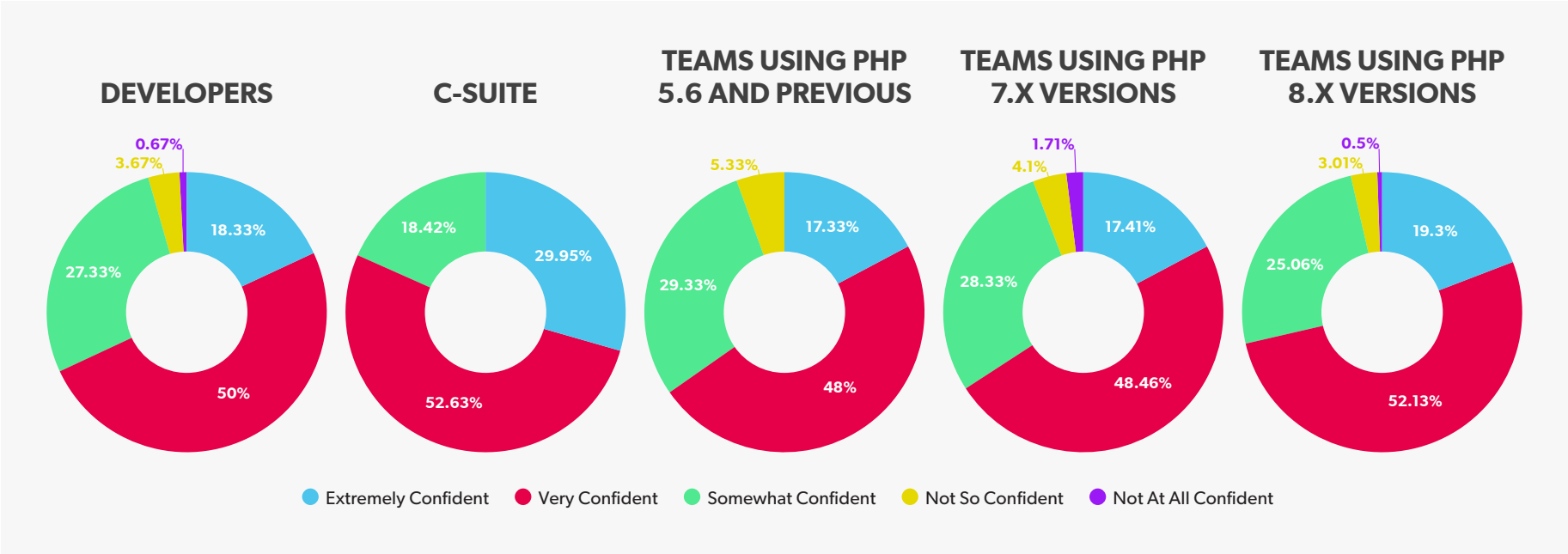


TEAMS WITH COMPLIANCE REQUIREMENTS



Next, we segmented the results by job responsibilities – specifically looking at hands-on development roles, vs. senior leadership roles. While the percentage of respondents for each category selecting Very Confident (50% vs. 52.63%) Somewhat Confident (27.33% vs. 28.95%) and Extremely Confident (18.33% vs. 18.42%) were relatively consistent, one area that showed a disconnect between roles were in negative perceptions of PHP application security. Developers, overall, were much more likely to select Not So Confident or Not at All Confident than their C-Suite counterparts (4.34% vs. 0%).

The last segment we looked at was security sentiment by PHP version adoption, comparing respondents using PHP 5.6 and previous version, PHP 7.x versions, and PHP 8.x versions. Looking at the results, teams using PHP 5.6 were far less confident in their overall PHP app security, while PHP 7.x and PHP 8.x users were much more confident.



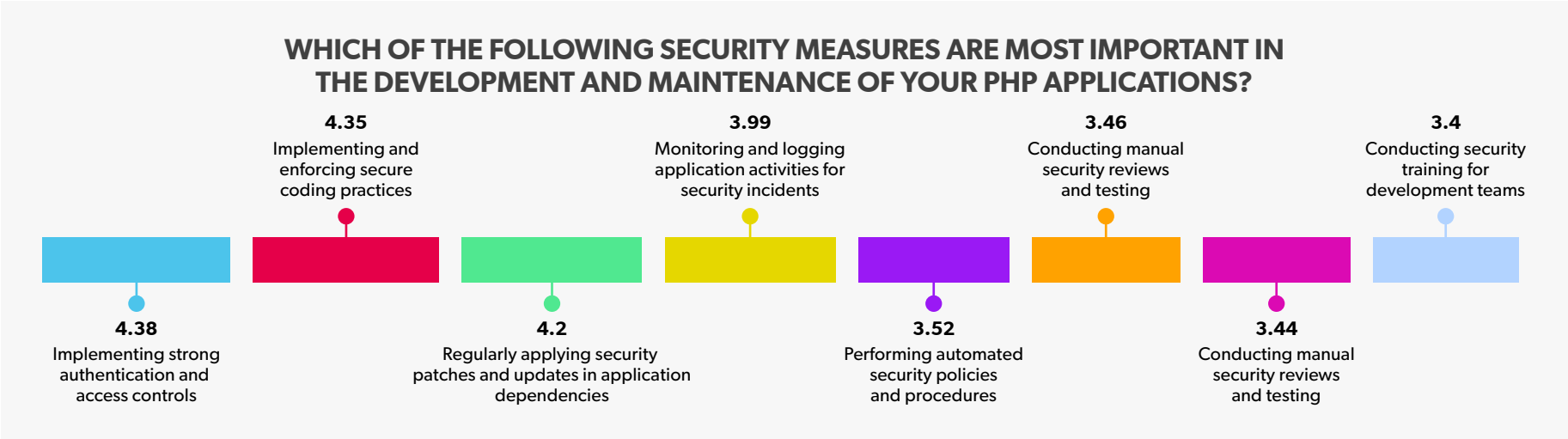
KEY TAKEAWAYS

We were surprised to see that the bulk of respondents are very confident in the security of their PHP applications. This speaks well to the improvements in the language and within its ecosystem over the past 5-10 years. We also felt that the respondents having less confidence in their applications that are deployed on version 5 releases was deserved; with the last PHP 5 release having been almost a decade ago, and security patching ending not long after, users should be looking to upgrade their applications in order to decrease their security exposure.

While confidence is still high for PHP 7 users, they, too, should be considering their options now that PHP 7 releases are no longer receiving security patches from the open source project.

TACTICAL SECURITY TRENDS

Next, we asked teams to share how they prioritize the measures they take to improve security in the development and maintenance of their PHP applications. Respondents were asked to rate each entry on a scale of 1-5, with 1 indicating that the tactic is not important, and 5 indicating the tactic is very important.



We then looked at the weighted average for each tactic, which you can see in the chart above. The consensus top three most important tactics for PHP teams were implementing strong authentication and access controls (4.38), implementing and enforcing secure coding practices (4.35), and regularly applying security patches and updates in application dependencies (4.20). Rounding out the top five were monitoring and logging application activities for security incidents (3.99) and performing automated security scanning and testing (3.52). Respondents rated conducting security training for development teams as the least important tactic (3.40).

KEY TAKEAWAYS

Identity and access management are a key concern when securing any application, and we are pleased to see so many respondents rating it as their chief security concern. Considering the PHP language does not provide any built-in mechanisms for achieving robust access control, choosing the right libraries and the right approaches are key architectural questions to answer.

Additionally, we were happy to see that implementing and enforcing secure coding practices is a top concern. We hope to ask what practices and standards in particular organizations are targeting when we do future iterations of this survey, as we are curious what language features are either aiding or obstructing the ability to secure applications. We also find it interesting that conducting security training is rated so low, considering training is necessary in order to effectively implement secure coding practices.

COMPLIANCE REQUIREMENTS

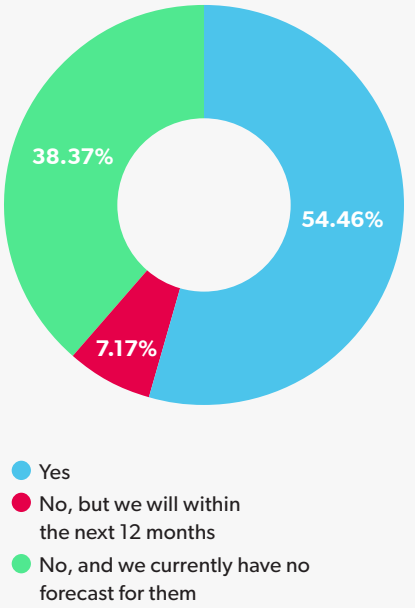
In a return question from our 2023 survey, we asked respondents to share whether they need to meet regulatory or industry compliance standards, and, if not currently needing to meet any compliance standards, whether they anticipate needing to meet any compliance standards within the next 12 months.

Our survey found that 54.46% of respondents noted they must meet regulatory or industry applications for their PHP applications, with 38.37% saying no, and that they don't currently foresee needing to meet any compliance standards. The remaining 7.17% indicated that they don't currently need to meet compliance standards but indicated that they will need to within the next 12 months.

Looking at these results year over year, we saw relatively little movement in teams needing to meet compliance (55.21% vs 54.46%). However, we did see a jump in those not needing to meet compliance standards and not expecting to need to meet them in the future (31.44% to 38.37%). Conversely, we saw teams noting they don't currently need to meet compliance standards but expect to in the next 12 months drop year over year (13.34% to 7.17%).

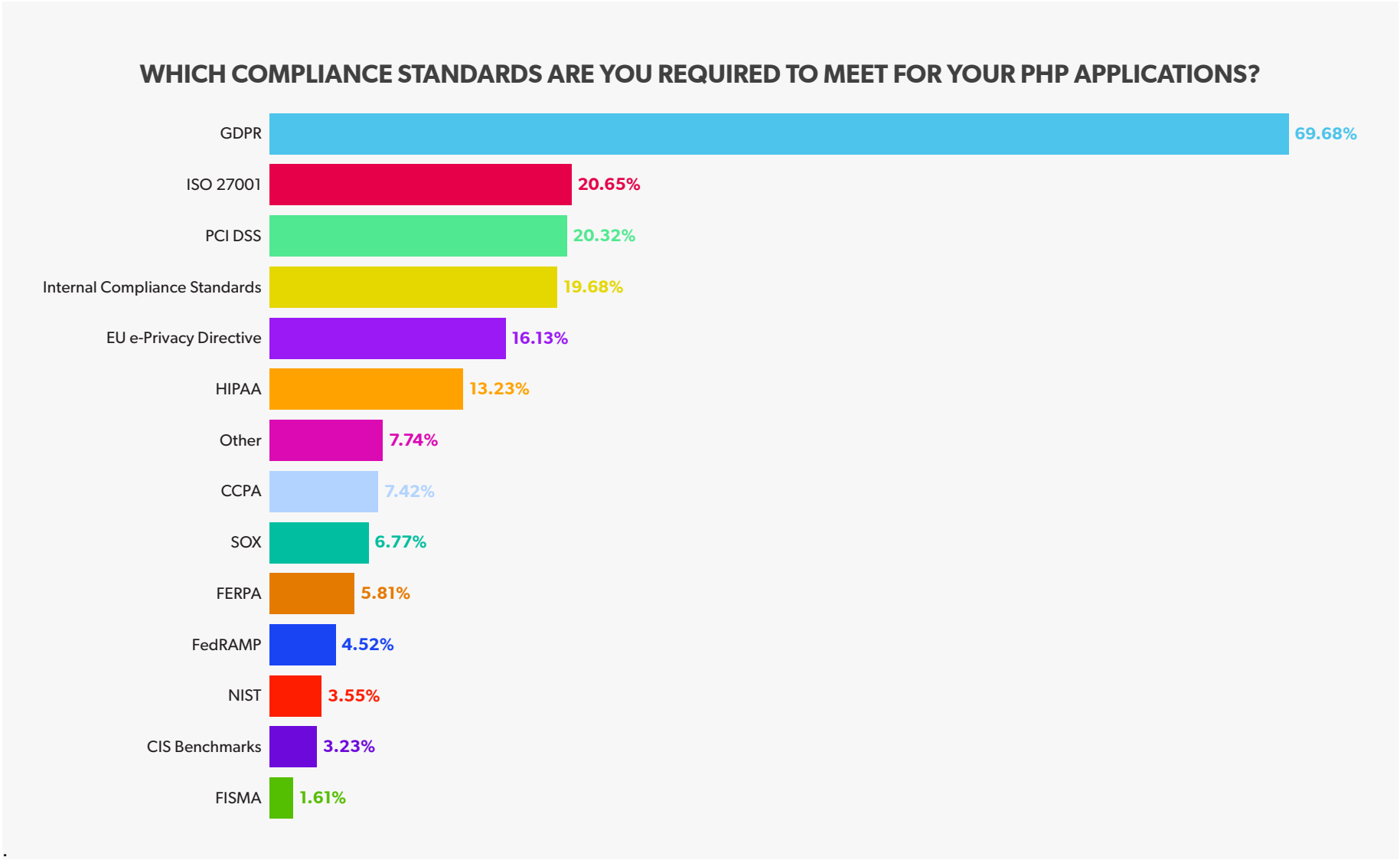
Looking at compliance requirement results by segment, our survey showed the top global regions needing to meet compliance standards for their PHP applications were North America (61.44%), Europe (60.35%), and United Kingdom (60%).

DO YOUR PHP APPLICATIONS HAVE REGULATORY OR INDUSTRY COMPLIANCE REQUIREMENTS?



COMPLIANCE STANDARDS

As a follow-up question, we asked those who indicated they need to meet compliance requirements to share the requirements that they adhere to for their PHP applications. Our survey found GDPR as the top selected option at 69.68%, followed by ISO 27001 at 20.65% and PCI DSS at 20.32%. Rounding out the top five, Internal compliance standards and EU e-Privacy Directive attracted 19.68% and 16.31%, respectively



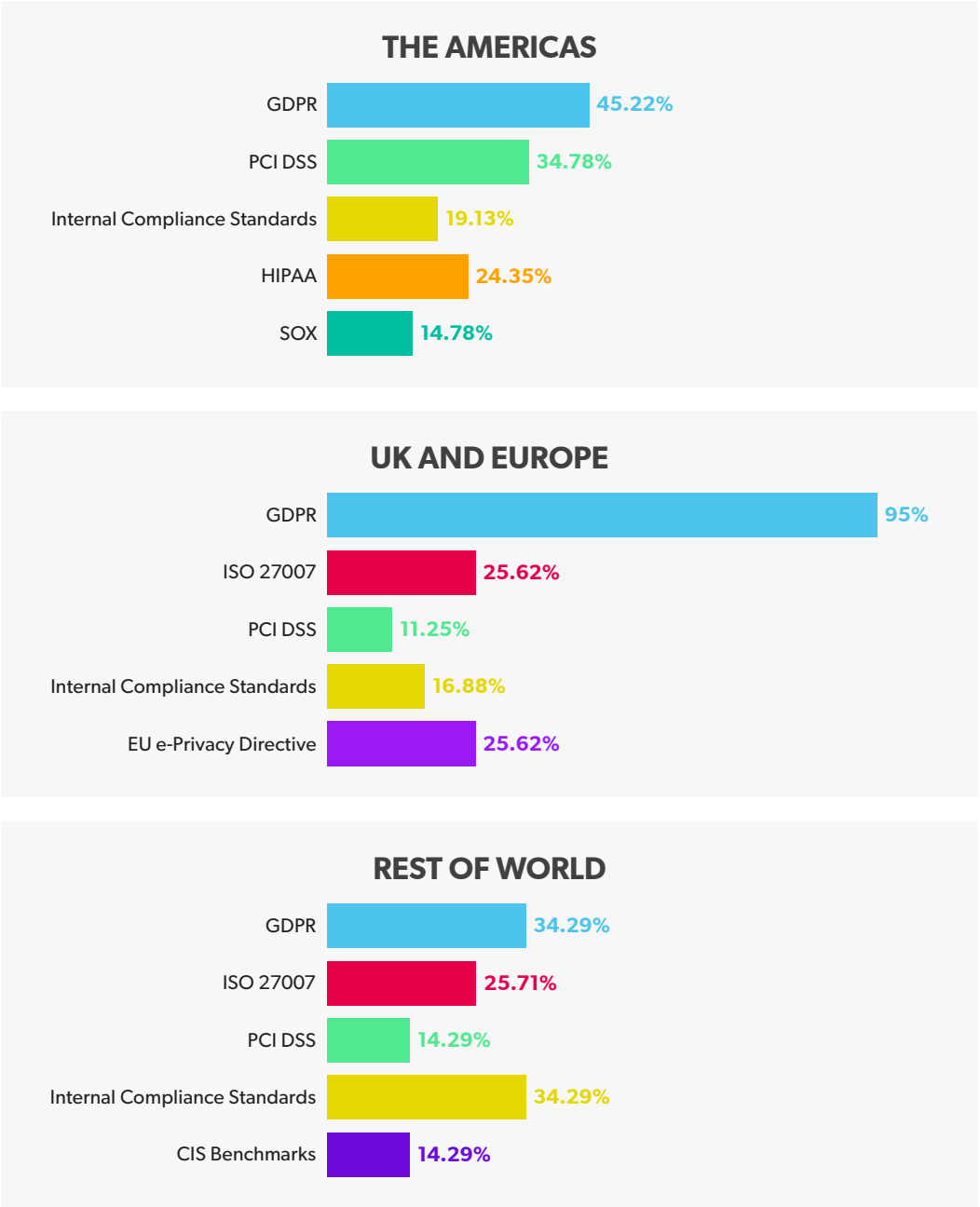
.

Not surprisingly, compliance requirements varied by region – with respondents in North, Central, and South America much less likely to select GDPR (45.22% vs. 95%) and EU e-Privacy Directive (5.22% vs 25.62%) than respondents from Europe and the UK.

KEY TAKEAWAYS

It's clear that PHP is powering global businesses. Government-led regulatory requirements such as GDPR, ISO 27001, the EU e-Privacy Directive, and HIPAA are governing how organizations deploy their applications. Additionally, money talks: PCI DSS, SOX, and NIST are required to interact with financial institutions, and compliance is a cost of doing business.

If your PHP application is working with any sensitive customer data, you have compliance requirements, and with the globally connected world of the internet, you're likely falling under many regulatory umbrellas.



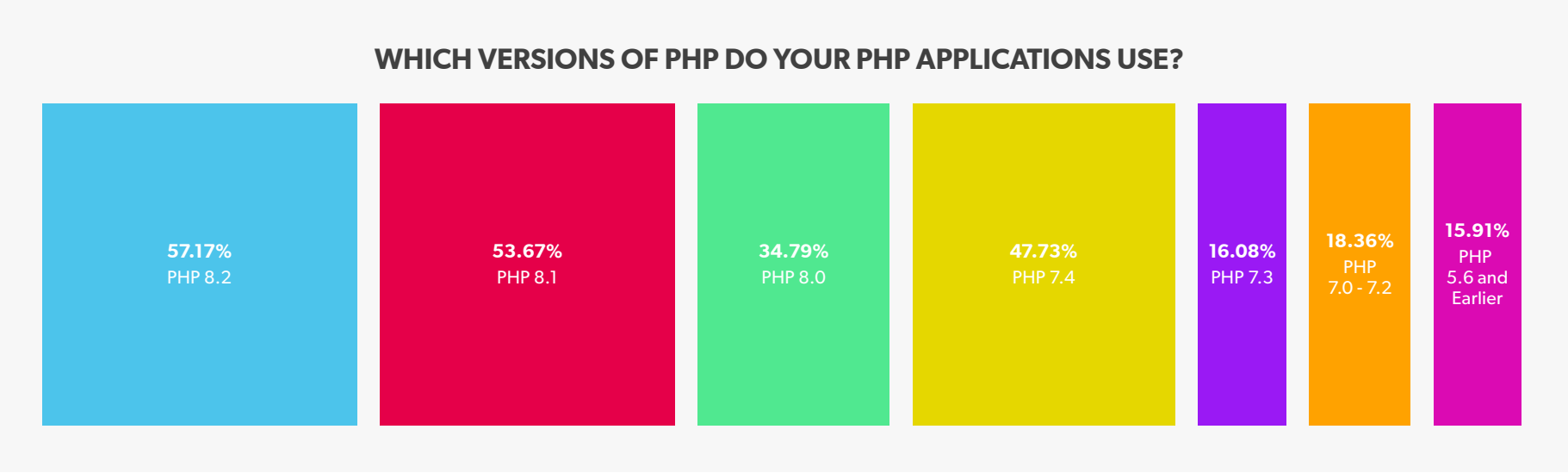
PHP VERSION ADOPTION AND MIGRATION TRENDS

Measuring PHP version adoption within the PHP ecosystem is the cornerstone of our annual PHP Landscape Report — and our 2024 report is no different. This year we again asked teams to share the PHP versions they use, and their plans around PHP upgrades or migrations. New this year, to better understand PHP migration trends, we asked users to share the PHP versions they migrated to and from. But before we get to that, let's dive in with the top PHP versions.

PHP VERSION ADOPTION

In this question, we asked teams to identify the PHP versions used in their PHP applications, with the option to select multiple PHP versions if applicable. Respondents reporting using 2.43 different PHP versions, on average (not accounting for answer selections that had multiple PHP versions included). This was lower than our 2023 survey, which found respondents selecting an average of 2.65 PHP versions.

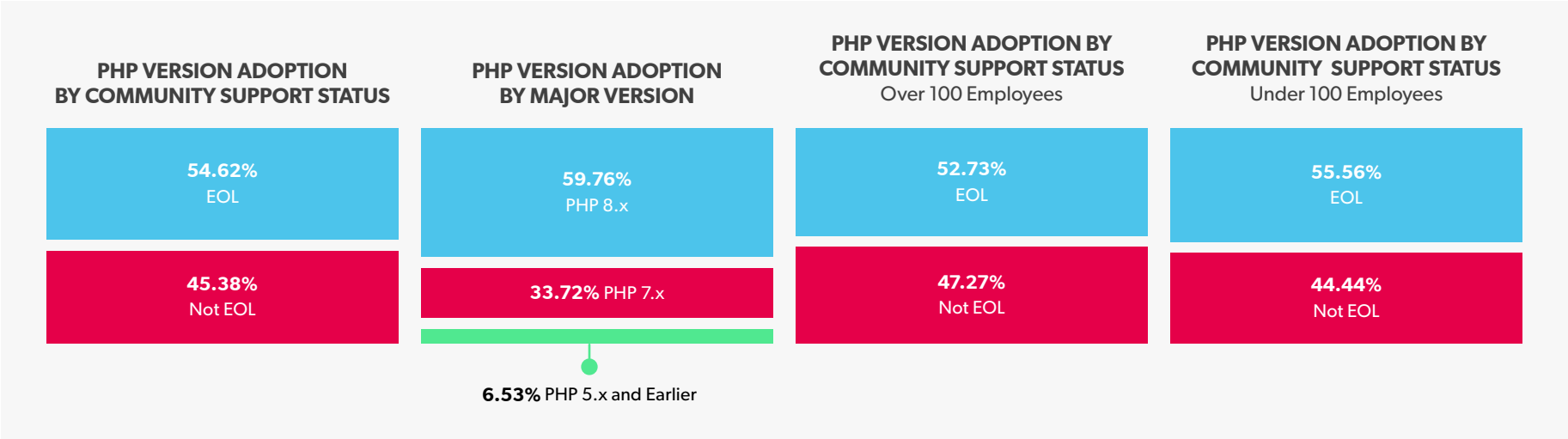
The top version selected was PHP 8.2, at 57.17% of responses, followed by PHP 8.1 at 53.67%, and PHP 7.4 at 47.73%. Next were PHP 8.0, PHP 7.0-7.2, and PHP 7.3 at 34.79%, 18.36%, and 16.08%, respectively. Last on the list were PHP versions 5.6 and earlier at 15.91% of respondents.



Overall, 54.55% of respondents noted using an end of life PHP version, marking a year over year decrease of 7.31%. Taking a closer look at end of life PHP version adoption year over year, PHP 7.4 usage dropped by 22.88%, PHP 7.3 dropped by 32.35%, and PHP 7.0-7.2 dropped by 45.08%.

Year over year, we saw adoption of PHP 8.x versions climb 56.58%, while adoption of PHP 7.x versions dropped 31.04% and adoption of PHP versions 5.6 and earlier dropped 49.44%.

Looking at EOL PHP version adoption by company size, we saw EOL PHP usage higher for companies with under 100 employees (55.56%), and lower for companies with over 100 employees (52.73%).



KEY TAKEAWAYS

We are currently in the second year where the only community-supported versions of PHP are in the version 8 series. When the last major release cycle happened (PHP 5 to PHP 7), it took until version 7.3 or 7.4 before we started observing significant migration off of PHP 5. With PHP 7 to PHP 8, we're seeing a faster uptick in adoption of the new version. While PHP 8 introduced a number of backwards incompatible changes, they were fewer and had less of an impact than we experienced with the release of PHP 7, which has led to greater success in adopting PHP 8.

As noted elsewhere in this report, we see a higher adoption of community-supported versions by larger organizations. We expect this is due to having more staff capable of maintaining applications and providing updates for compatibility.

Finally, we would like to note that among teams who indicated a lack of confidence in the security of their PHP applications, 72% of those respondents also noted usage of end-of-life versions of PHP. Clearly, keeping applications on supported PHP versions goes hand in hand with good security practices.

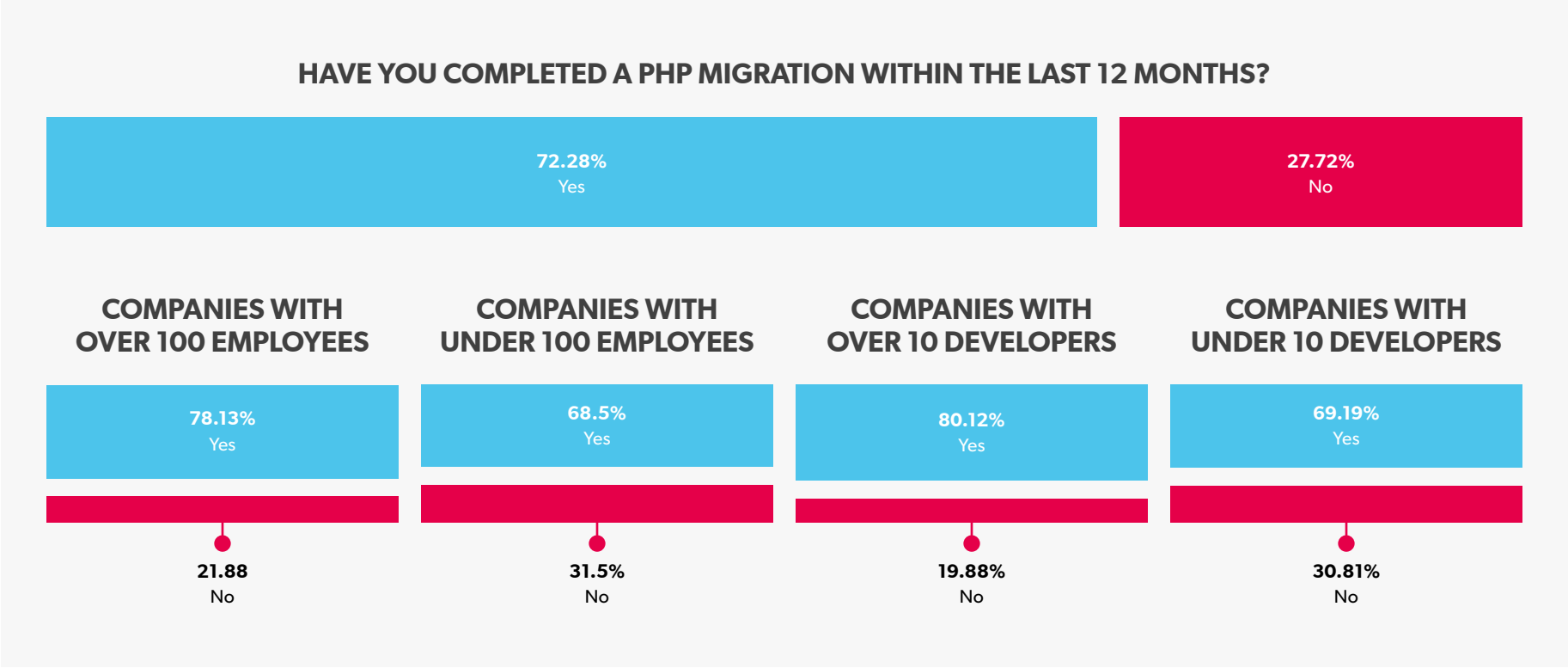
PHP UPGRADE AND MIGRATION TRENDS

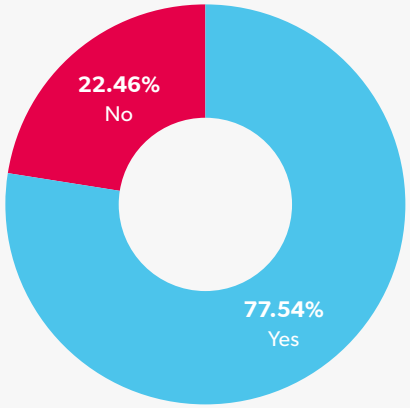
In our next set of questions, we asked teams to share details on their PHP migration efforts, including whether or not they've completed a migration within the last 12 months, and, if so, which PHP versions they migrated from and to. We also asked teams to share if they have plans to migrate PHP versions within the next 12 months. Finally, we asked teams to share the biggest obstacles to their PHP migrations.

PHP Migrations Within Last 12 Months

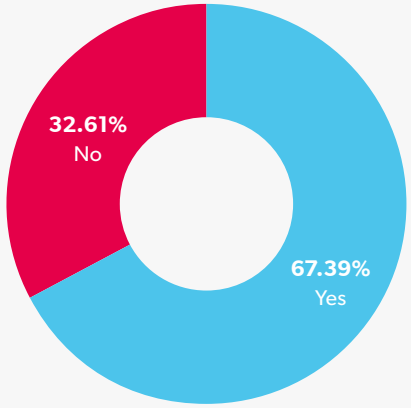
In our first question for this section, a new question in this year's survey. We asked teams to share whether or not they have completed a PHP migration within the last 12 months. Our survey found that the vast majority of respondents (72.28%) had completed a PHP migration during that period.

Looking at migration activity in large companies versus small companies, we saw large companies much more likely to have completed a PHP migration within the last 12 months when compared to small companies (78.28% vs 68.50%).





COMPANIES USING CONTAINER/ OCHESTRATION TECHNOLOGES



COMPANIES NOT USING CONTAINER/ OCHESTRATION TECHNOLOGES

Digging deeper, we looked at results by development team size and found even more pronounced separation. Respondents who indicated they had fewer than 10 developers on their development team were far less likely to have performed a migration in the last 12 months (69.19% vs. 80.12%).

Looking at migration activity for companies who noted usage of containerization and orchestration technologies, we noted that they were more likely to have completed a migration than those not using those technologies.

KEY TAKEAWAYS

As noted in the commentary, respondents with small teams are less likely to perform and complete PHP migrations; this was mirrored in the results of teams who have not adopted containers or orchestration technologies. Migrations not only involve ensuring that the code works correctly on the new version, but also involve infrastructure provisioning and configuration — which containers and orchestration can help achieve in testable and repeatable ways. Often, changing a PHP version in such systems is a small change in configuration, and the results can then be tested in existing CI/CD pipelines.

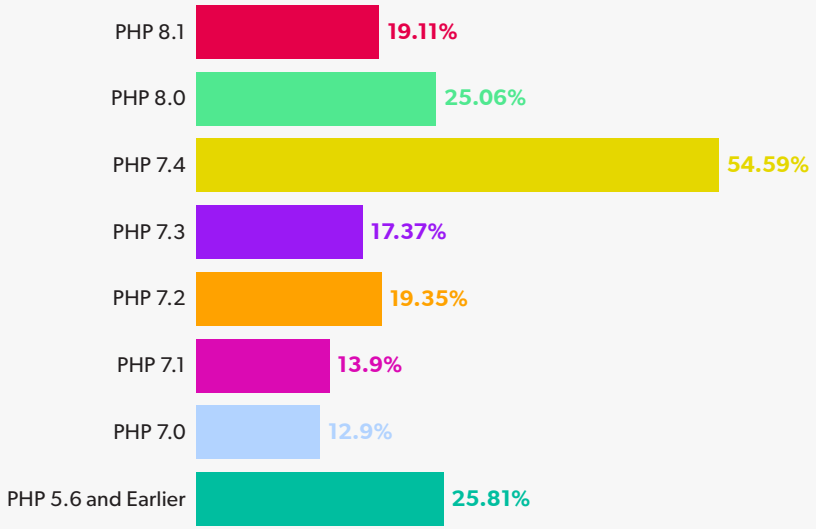
We strongly feel that container adoption and/or usage of orchestration tooling helps organizations be more successful in PHP migrations, and also contribute to greater application security practices.

PHP MIGRATION TRENDS BY VERSION

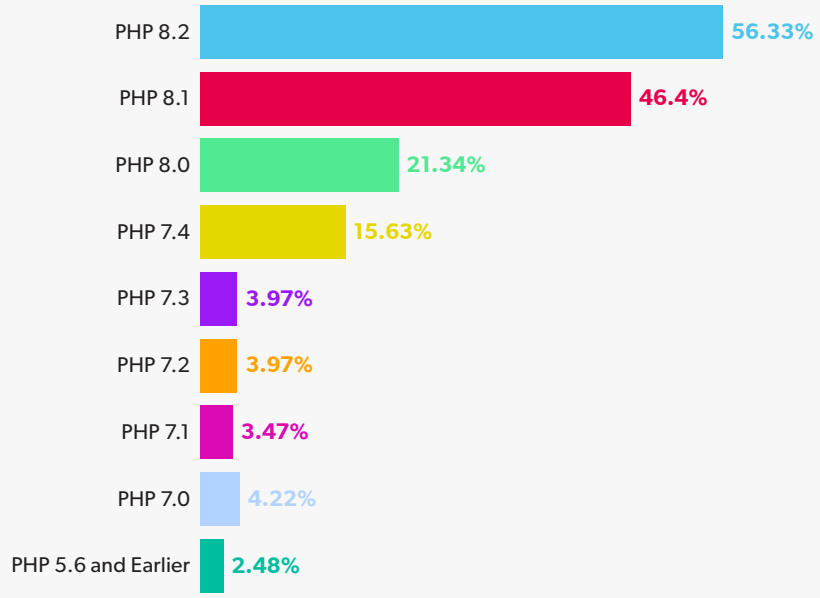
New to the survey this year, we asked teams who indicated they had performed a migration in the last 12 months to select which version(s) they had migrated from and to. Looking at our results, the top PHP versions respondents migrated off of were PHP 7.4 (54.59%), PHP 5.6 or older (25.81%), and PHP 8.0 (25.06%). Rounding out the top five were PHP 7.2 and PHP 7.3, at 19.35% and 17.37%, respectively.

The top PHP migration destinations for those versions were all PHP 8.x versions, with PHP 8.2 as the top choice at 56.33%. PHP 8.1 and PHP 8.0 followed at 46.40% and 21.34% respectively. Lastly, 15.63% of respondents noted that they had performed a migration to PHP 7.4 in the last 12 months.

WHICH PHP VERSION(S) DID YOU MIGRATE OFF OF?



WHICH PHP VERSION(S) DID YOU MIGRATE TO?

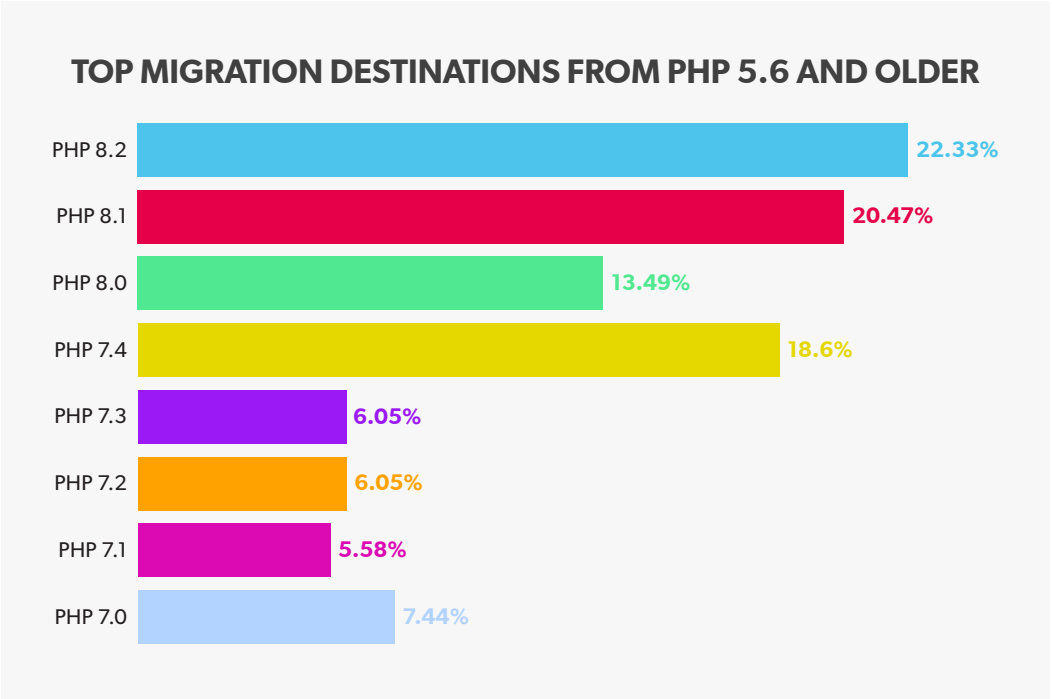


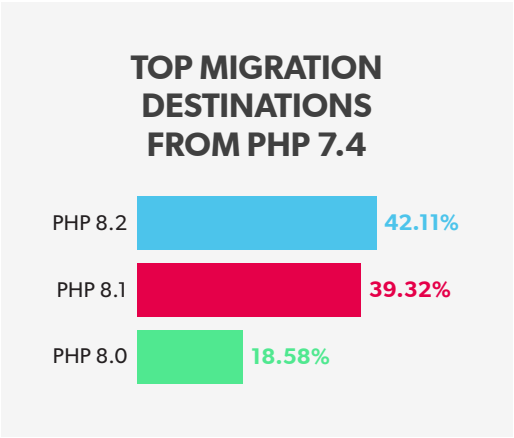
Looking at migration destinations for teams who indicated they migrated off of PHP versions 5.6 and Older, we found the top three migration destinations as PHP 8.2 (22.23%), PHP 8.1 (20.24%), and PHP 7.4 (18.6%). For teams that indicated they migrated off of PHP 7.4, the top migration destinations were PHP 8.2 (42.11%), PHP 8.1 (39.32%), and PHP 8.0 (18.58%).

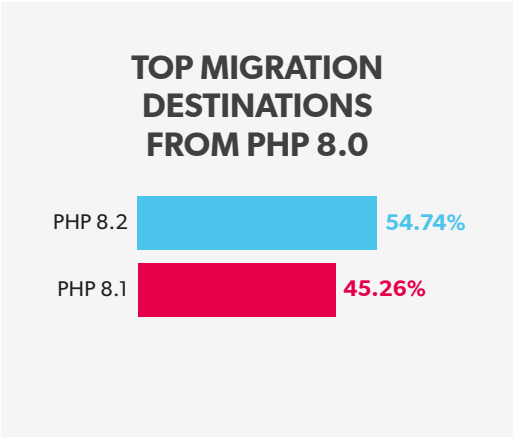
Lastly, looking at teams who indicated migrating off PHP 8.0, the top choice was PHP 8.2 at 54.74%, followed by PHP 8.1 at 45.26%. It is worth noting, however, that PHP 8.3 was not included as a potential option because it had not been released by the launch of our survey.

KEY TAKEAWAYS

While it should be the goal of most organizations deploying PHP applications to be on versions of PHP under current community support, we also note that for those on older PHP 7 versions or PHP 5 versions, the logical waypoint for migrations is to get to PHP 7.4. A number of long-term support offerings exist for PHP 7.4, including from Zend by Perforce, and getting to this version and addressing language deprecations will set your business up for success when preparing for a migration to PHP 8.





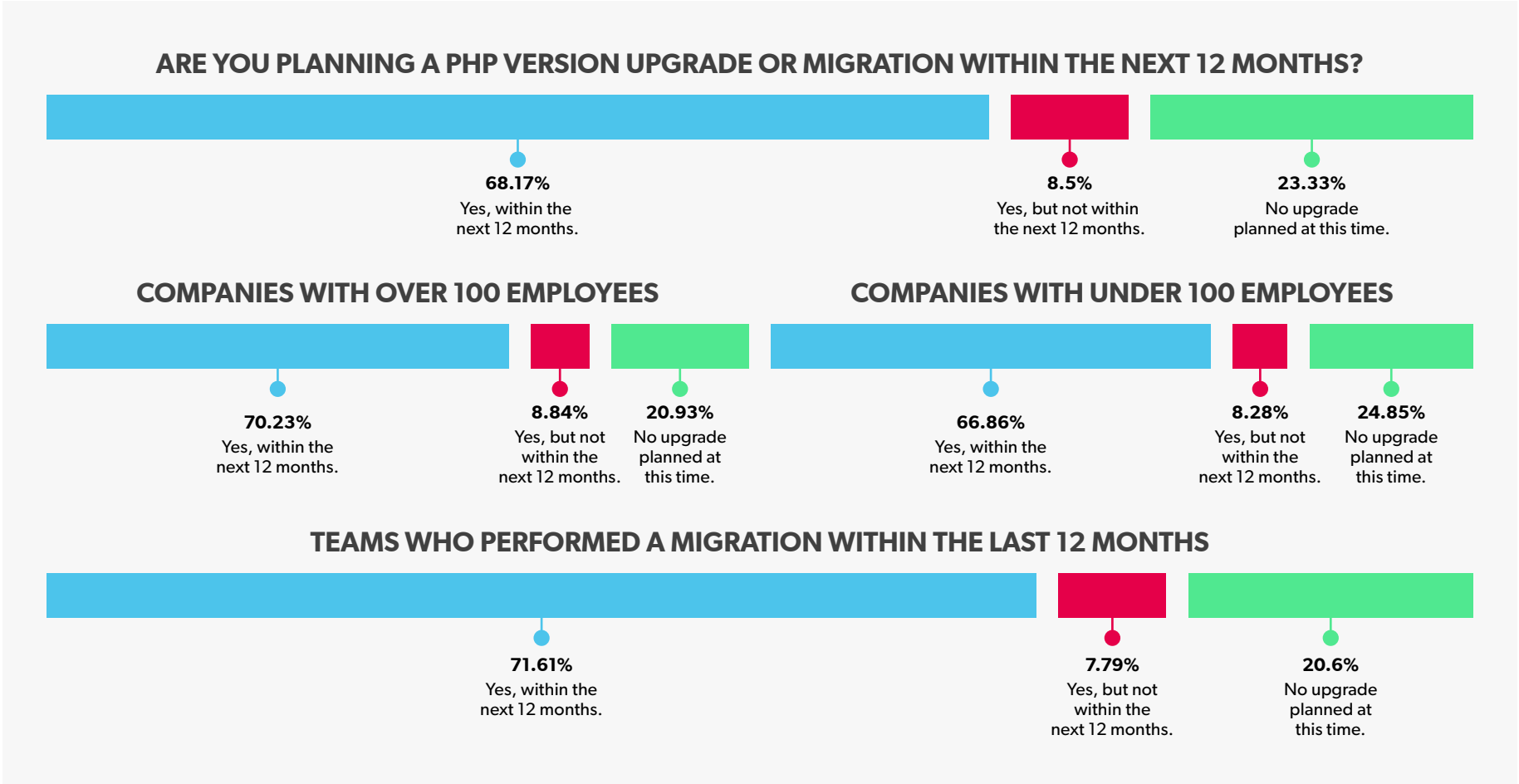


Forward-Looking PHP Upgrade and Migration Trends

In our next question, we asked respondents to share their plans regarding PHP version upgrades and migrations within the next 12 months. 68.17% of respondents indicated that they have plans to upgrade within the next 12 months. 23.33% of respondents noted that they have no upgrade planned at this time.

Looking at upgrade/migration plans by company size, we noted two minor shifts between large and small companies, namely that large companies were more likely to have migration plans than small companies, and less likely to have no upgrades planned.

Next, we looked at migration plans for respondents that noted they had performed a migration within the last 12 months. We found that, even for teams that had recently performed a migration, 71.61% have a migration planned within the next year. A further 7.79% indicated that they have a migration planned, but not within the next 12 months.



MIGRATION PAIN POINTS

In our next question, we asked teams to share the most time-consuming component of their last PHP upgrade – with responses limited to a single selection per respondent. The top two time-consuming elements of migrations noted by our respondents were Refactoring, at 37.31%, followed by Testing at 37.12%. Trailing significantly behind our top two choices, Infrastructure Provisioning, Planning, and Compliance Renewals, were the next most popular choices at 9.23%, 9.04% and 3.65%, respectively. Among those who selected "Other," dependencies were noted as the most time-consuming element of their last PHP upgrade.

While Refactoring and Testing were neck and neck as the top pain point for all respondents, when segmenting responses by teams who had migrated off of PHP 7.x versions, we noted that Refactoring (42.75%) was selected at a higher rate than Testing (35.69%), climbing above the 37.31% noted in the non-segmented results.

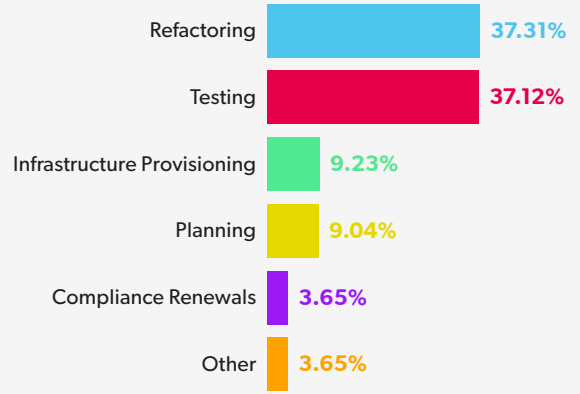
KEY TAKEAWAYS

While the results for planned upgrades/migrations do not change much year over year, one thing to note is that even with companies that have completed a migration in the past year, another is planned for the coming year. In working with our customers, we've noted that migrations often are staged: companies need to go to an intermediary version before getting to their target version, and each stage can be a months or years long project.

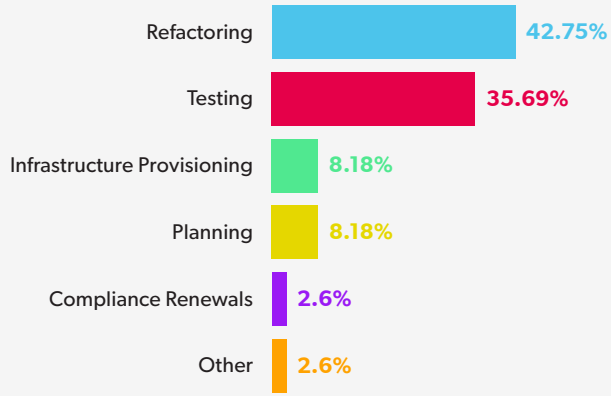
Additionally, we note that this churn is largely a by-product of the PHP release cycle itself. A little over a 10 years ago, the project adopted a regular release cadence that includes a yearly feature release, with each such release receiving 2 years of active support with 1 additional year of security-only support. As such, this means migrations every 2-3 years at a minimum.

When it comes to the details of migration, refactoring was rated as the biggest pain point. Again, this can be attributed to the pace of change in the language: every feature release can potentially introduce deprecations, and if it represents a new major release, backwards incompatible APIs. These can require refactoring your code in order to work under the new (or a future) version. Additionally, some features, when adopted, can increase code readability or predictability (as examples, the type improvements from PHP 7 and PHP 8; read-only properties and classes in PHP 8; and more); refactoring your code to take advantage of these may give your team greater ability to produce new features in the future.

WHAT WAS THE MOST TIME-COMSUMING COMPONENT OF YOUR LAST PHP UPGRADE?



PAIN POINTS FOR TEAMS WHO MIGRATED OFF PHP 7.4



DEVELOPMENT PRIORITIES

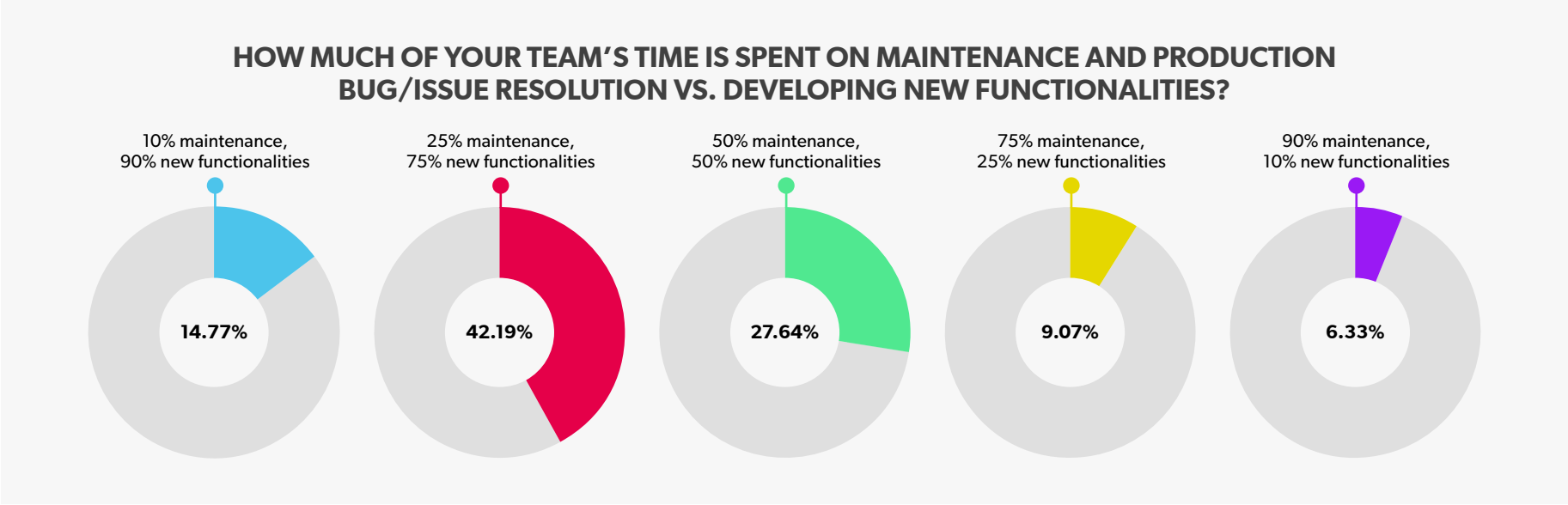
In the penultimate section of our 2024 PHP Landscape Report, we asked teams to share the focus areas they have today, and the development priorities they have set for the rest of 2024.

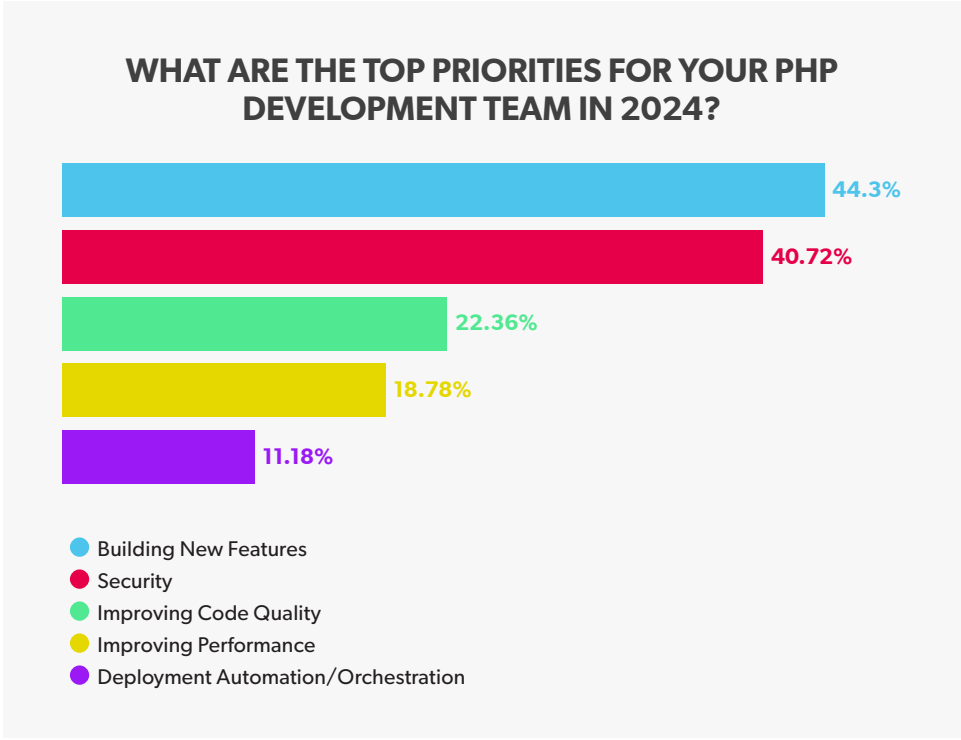
MAINTENANCE VS. FEATURE DEVELOPMENT

In our first question for this section, we asked respondents to share how they prioritize maintenance of their PHP applications, including bug/issue resolution, vs. the development of new functionalities. The top selection in this year's survey, with 42.19% of responses was "25% maintenance, 75% new functionalities". Next was a 50/50 split, at 27.64%. In third, 10%/90% split attracted 14.77% of responses. Overall, our survey found 84.6% of respondents commit at least half of their time to developing new features.

KEY TAKEAWAYS

With 7 out of 8 respondents indicating they commit at least half their time to new features, businesses can expect that PHP developers are producing business value for them, first and foremost.





2024 DEVELOPMENT PRIORITIES

In our next question, we asked teams to share their 2024 development priorities. Users were asked to rate each category on a scale of one to five, with one indicating the lowest level of priority, and five indicating the highest. Echoing the results from our previous question, we found 44.30% of respondents setting the highest priority on Building new features. In second, trailing our first place option by 3.58%, was Security at 40.72%. Improving code quality, Improving performance, and Deployment automation / orchestration represented 22.36%, 18.78%, and 11.18% of respondents, respectively. Looking at development priorities by company size, we didn't note any substantial shifts.

KEY TAKEAWAYS

With new feature development and security being on par as priorities, the question we have is whether respondents were thinking about security-oriented features (such as identity and access management), or if they are expecting security audits, and resulting mitigations, of their applications.

THE STATE OF PHP IN 2024

In our final section of our survey, we asked respondents to share what they consider to be the most important components of PHP as a language, the challenges they encounter in working with PHP, and their views on the state of PHP today.

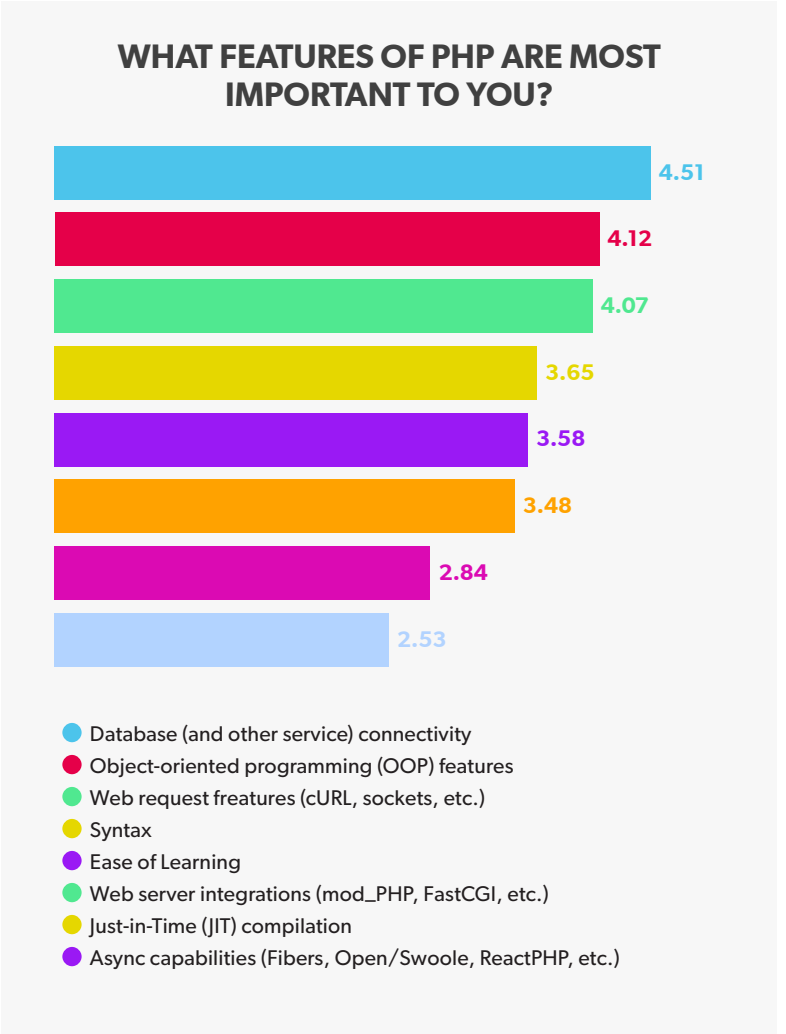
MOST IMPORTANT PHP FEATURES

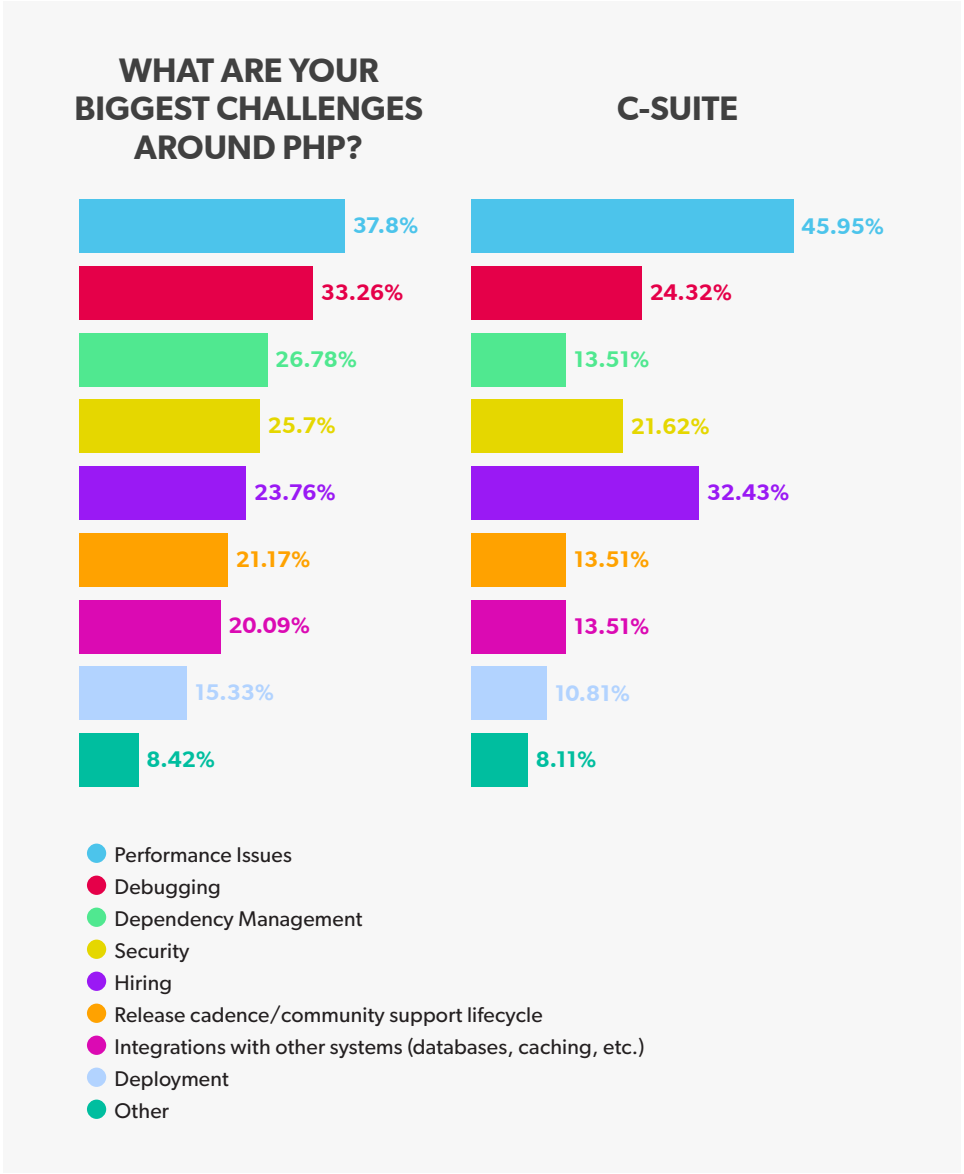
In our first question for this section, we asked respondents to rank in order of importance a series of PHP features and qualities ranging from Database connectivity to Syntax. The top feature again in this year's survey was Database (and other service) connectivity, which attracted a weighted average of 4.51. Next was object-oriented programming features at 4.12, followed by Web request features (cURL, sockets, etc.) at 4.07. Syntax and Ease of Learning were next at 3.65, and 3.58, respectively. Looking at results year over year, we noticed a few movements. Web request features dropped from the 2nd most important feature to 3rd, replaced by OOP features. We also noted that Syntax climbed from 6th to 4th, while Web server integrations fell from 4th to 6th.

KEY TAKEAWAYS

As usual, the ability to integrate with a variety of services and systems is a leading feature of PHP. We are pleased to see that respondents are increasingly finding the language features and syntax an important part of choosing the language, as it demonstrates that the change in focus by the project has benefited users.

Finally, we note that the drop in importance of web server integrations is likely due to these being a given, particularly with the two most popular servers used when deploying PHP (Apache HTTPD and nginx).





BIGGEST PHP CHALLENGES

Next, we asked teams to weigh in on the biggest challenges they face in working with PHP, with the option to select multiple challenges as applicable. Our survey found the top challenge for respondents to be Performance issues, at 37.8%, followed by Debugging at 33.26%. Dependency management, Security, and Hiring were the next most common challenges at 26.78%, 25.70%, and 23.76%, respectively.

While the top two challenges for teams stayed consistent year over year, Dependency management jumped from 5th to 3rd, while hiring dropped from 4th to 5th. Integrations with other systems dropped from 3rd in 2023, to 7th in 2024.

When looking that top challenges for executive level positions, we found that Hiring jumped nearly 10%, and from the 5th most common challenge to 2nd.

KEY TAKEAWAYS

We note that dependency management jumped two positions this year. While dependency management is a largely solved problem due to the ubiquity of the Composer dependency manager, we wonder if the challenge posed here is more around keeping up-to-date with secure versions of dependencies.

We are also surprised by the increased challenge in hiring. We are curious if this is indicative of a decrease in PHP developers, or if the demand for PHP developers has increased in the past year.

THE STATE OF PHP

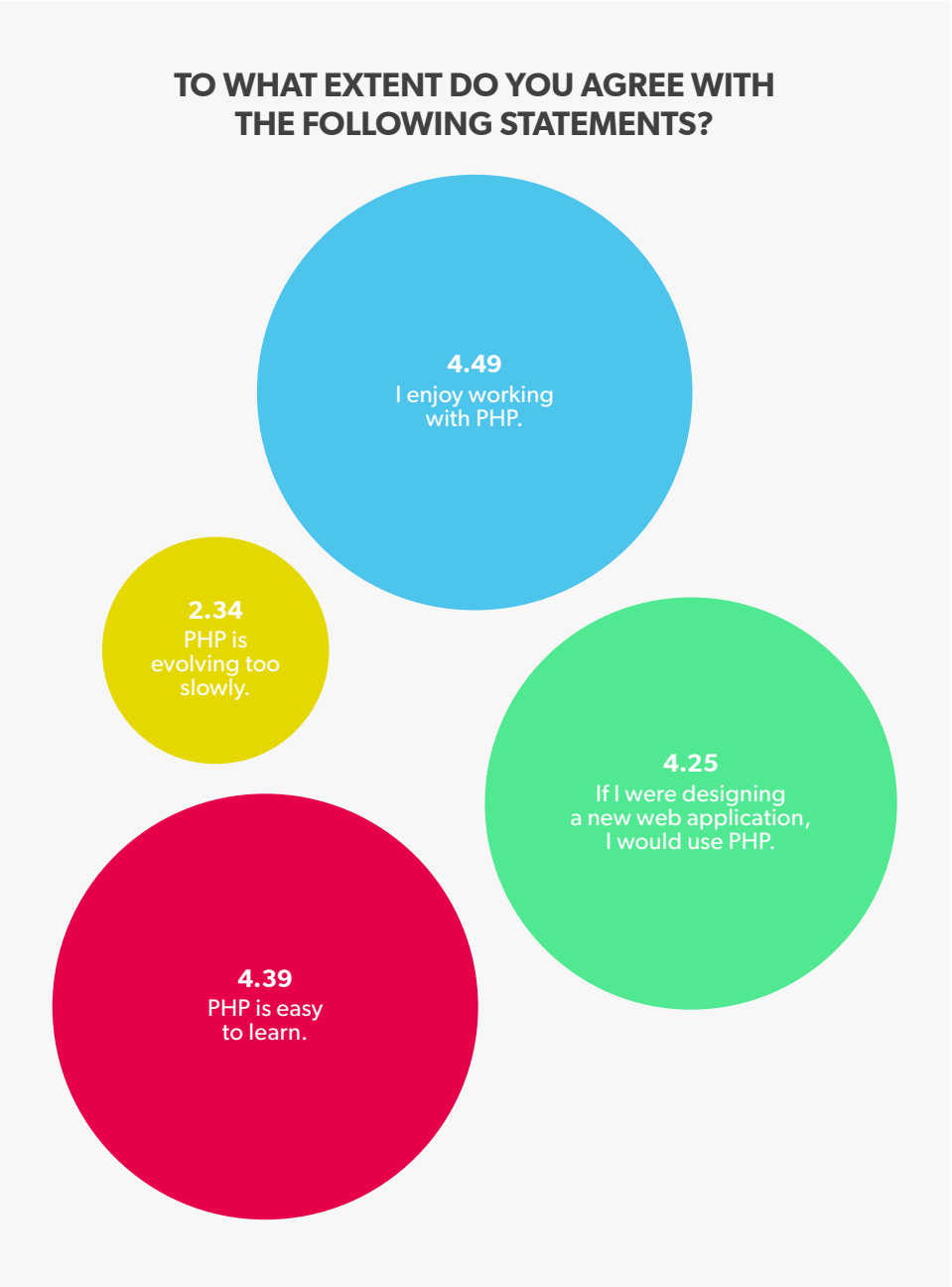
In our final question for the 2024 PHP Landscape Survey, we asked teams to share their level of agreement with four prompts regarding PHP. Respondents were asked to rank each category from one to five, with one representing the lowest level of agreement and five representing the highest level of agreement.

Our survey found that respondents strongly agreed with the first three prompts, in that they enjoy working with PHP, that it is easy to learn, and that if they were developing a new web app, they would use PHP. Most respondents disagreed with the statement that PHP is evolving too slowly.

Looking at these results year over year, we found that these opinions remained consistent. The statements "I enjoy working with PHP" and "PHP is easy to learn" increased in agreement year over year, while more people disagreed with the statement "PHP is evolving too slowly."

KEY TAKEAWAYS

The PHP language has been receiving a plethora of useful features over the past few years, many of which make it easier to write maintainable and secure software. The annual drop of new features keeps many developers excited about the language, and invested in learning and increasing their efficiency in developing business critical software.



Final Thoughts

Whether your business has a web presence managed by a CMS, or exposes APIs for business-to-business enablement, you likely have PHP somewhere in the stack. We have observed it in wide use across all industries, from automotive and industrial companies to educational institutions and the public sector. Its ability to glue together data and services from a variety of sources keeps it relevant, and we continue to see it evolve to suit complex needs.

With PHP 8, we've seen increased stability in the language, and a number of forward-thinking features such as the Just-In-Time compiler and Fibers, which should enable new use cases and paradigms in the future. All of this has been done in a way that allows PHP to continue shining where it always has: enabling scalability, which in turn helps businesses control their costs.

What will another year bring to PHP? Let's find out together!

ABOUT ZEND

Zend by Perforce helps organizations use enterprise PHP to build innovative web and mobile solutions and modernize existing applications. Used by multiple Fortune 100 companies, our proven enterprise PHP offerings include secure, fully-supported PHP runtimes, software infrastructure, tools, training, and enterprise long-term support for PHP 7.2, 7.3, 7.4, and 8.0. For more information, visit www.zend.com today.

Learn More About Our PHP LTS Options

EXPLORE LTS OPTIONS

Learn More About Our Professional Services

SEE WHAT WE OFFER