



2024 Java Developer Productivity Report

Java Development Trends
+ Analysis



Foreword

Dear Colleagues,

Welcome to the 2024 Java Developer Productivity Report. If this is your first time reading an edition of this annual report, we're glad you're here. We hope the proceeding pages provide valuable benchmarks on the challenges your Java development peers are facing and the Java tech stack they're using to solve them.

If you've been following the Java Developer Productivity Report for each of the dozen years it has been produced: welcome back. The topics may have changed over time, but our commitment to helping you solve your toughest Java development challenges has not.

In 440 responses to our survey, three key themes rose to the top. First, Java is here to stay: 60% of respondents say their companies plan to add Java developers in the coming year, and 42% plan to increase their Java development tool budget. These investments in tools and talent are testament to Java's role as the stable backbone of enterprise applications.

Secondly, cloud is a necessity to remain agile in today's business environment. Only 13% of respondents reported that they don't use any cloud providers. But the investment in cloud only makes sense if your tooling and processes can operate at the speed of cloud; a quick, iterative DevOps loop is critical. Without it, cloud deployments become the equivalent of driving a Ferrari to the grocery store. But results show that some companies may be struggling to keep pace, with friction surrounding performance and deployment speeds.

The final theme is perhaps the most important: developer productivity teams are paving the path forward. These teams investigate solutions and test workflows so developers can concentrate on writing code—and pay dividends as development environments become increasingly more complex. This is the first year we've tracked the incidence of dedicated productivity teams, and at 42% we only expect that number to skyrocket.

We hope the data presented in this report helps make your application development decisions easier by providing clear benchmarks on Java trends—as well as insights on how innovative companies can leverage these trends to gain a competitive edge.

Enjoy the report,

Rod Cope | *Chief Technology Officer, Perforce*



Rod Cope

Chief Technology Officer,
Perforce

Table of Contents

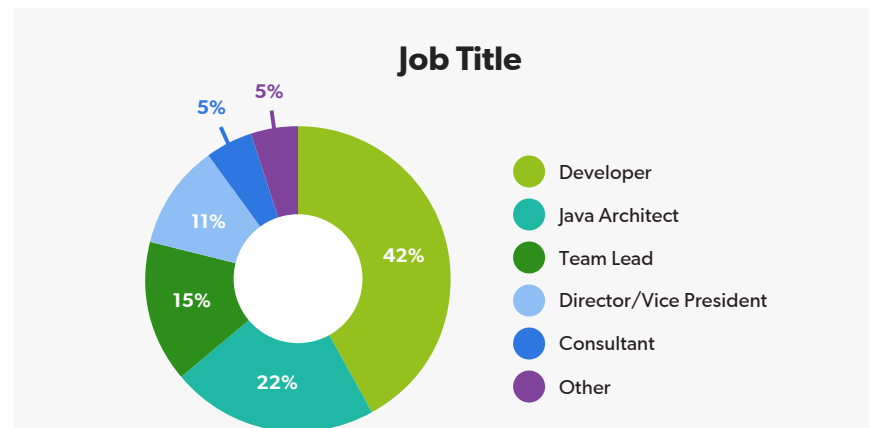
About the Survey.....	04
Investments in Java.....	05
Java Language + Technology Trends	06
Microservices + Cloud	10
Productivity Trends	13
Final Thoughts	15

About the Survey

The 2024 Java Developer Productivity Report is based on an anonymous survey conducted between November 2023 and January 2024. The survey received a total of 440 responses from 72 countries, with the majority of responses coming from the U.S. (27%), China (16%), India (6%), Germany (4%), and Italy (3%).

JOB TITLE

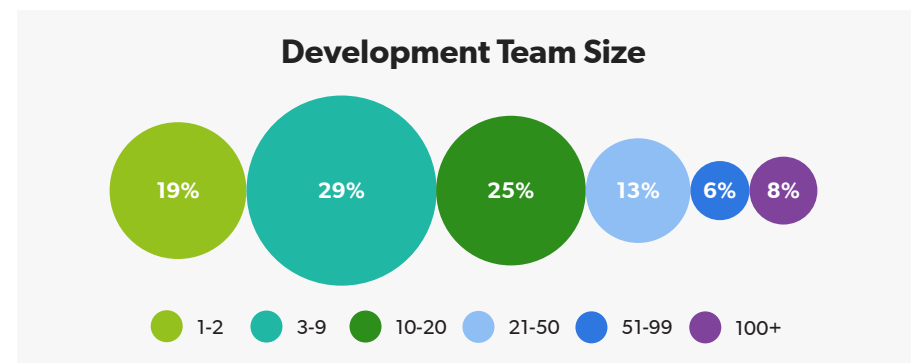
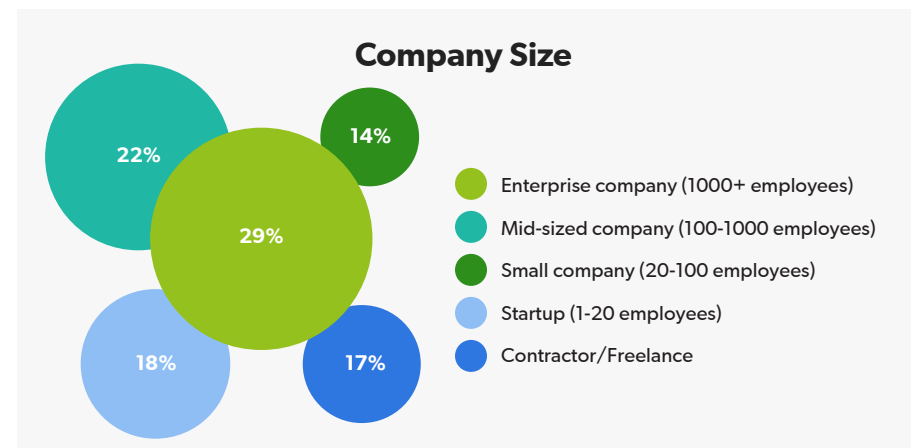
While most respondents listed their role as Java developer (42%) or Java architect (22%), this year's survey saw more respondents in leadership roles compared to 2023, including team lead (15%) and director/vice president (11%).



TEAM AND COMPANY SIZE

Respondents represented a wide range of company sizes from enterprises and midsize organizations to startups and freelancers. Company size spread was mixed, with 29% from companies with 1,000+ employees, 22% from companies with 100-1,000 employees, 14% from companies with 20-100 employees, 18% from companies with 1-20 employees, and 17% describing their roles as contractor or freelance.

This year's survey also saw an increase in responses from those on Java development teams with 10 or more developers over 2023. That said, the greatest number of responses came from those on teams of 3-9 developers (29%); that makes sense because most scrum teams cap at nine developers.

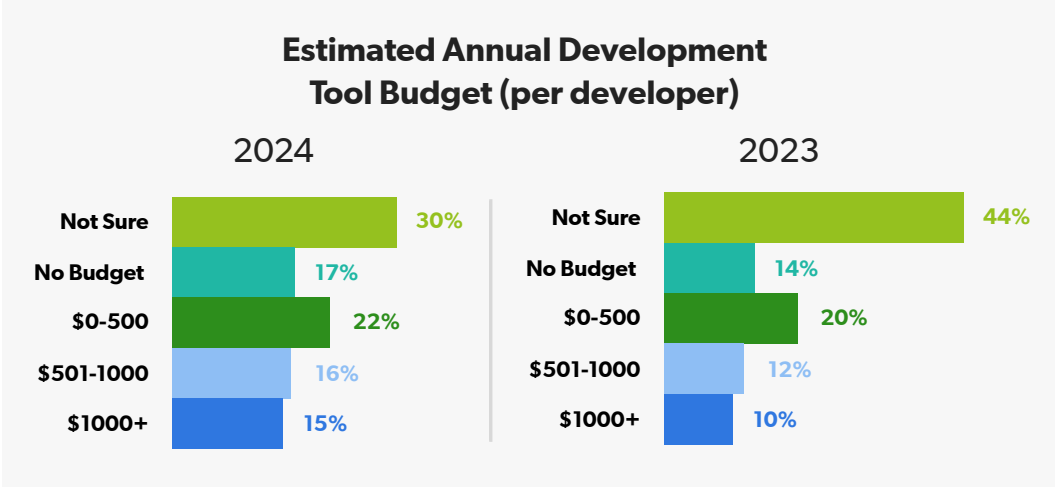
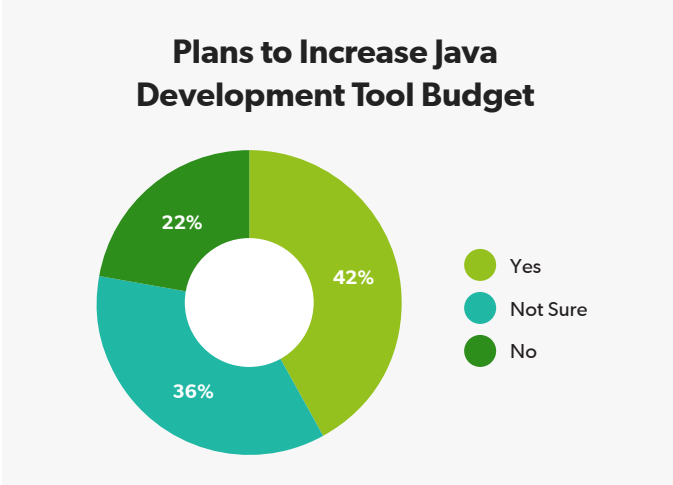
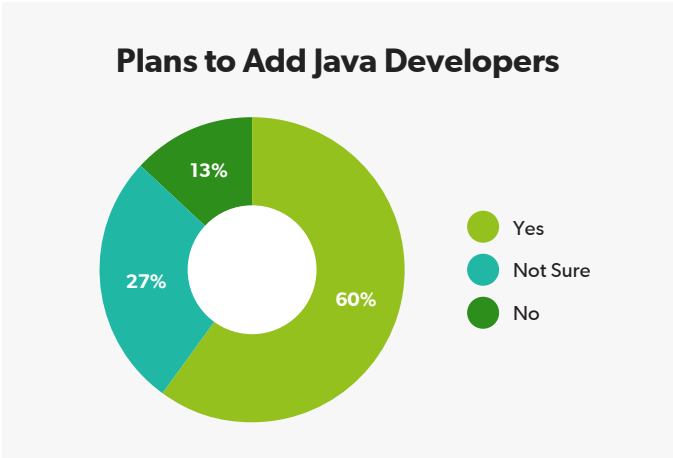


Investments in Java

Overall, results show that investments in Java tools and talent are on the rise. 60% of respondents said that their companies planned to add additional Java developers in the coming year and only 13% said they did not plan to add Java developers; an additional 27% said they weren't sure.

Developer tool budgets appear to be holding steady as companies face economic headwinds. 42% plan to increase their tool budget, 22% of respondents do not plan to increase their tool budget, and 36% weren't sure. Moreover, 31% said their annual tool budget (per developer) was \$500 or more. This is an increase from 2023, where only 22% of respondents said their tool budget was \$500 or more.

In conjunction, these numbers send a strong message that Java is here to stay as a core part of enterprise applications. The fact that companies are very deeply entrenched with large Java applications will continue to be a driving force for hiring Java developers throughout the developer ecosystem.



Java Language + Technology Trends

In this section, find out what your peers are using in their Java environments. While some results—like the most popular application servers or Java frameworks—are status quo, others show clear trend changes for 2024.

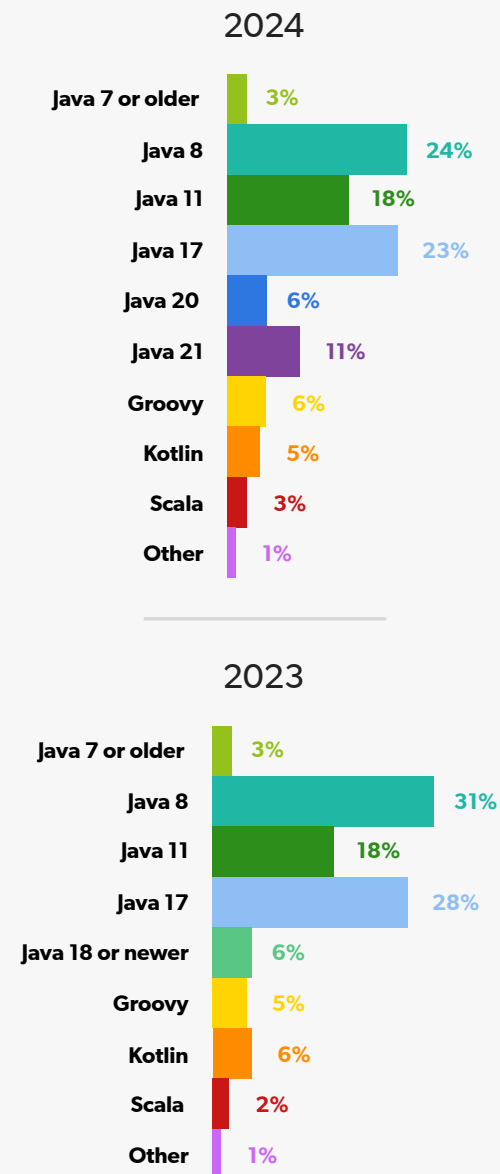
LONG-TERM SUPPORT AND SECURITY

Respondents were asked what programming language or languages they were using in their main application, and the results were polarized. 11% of respondents said they had already upgraded to Java 21 (released in September 2023).

Conversely, 24% of respondents said they were using Java 8 and another 18% respondents said they were using Java 11. Oracle discontinued Premier Support for Java 8 in March 2022 and Java 11 in September 2023. Thus, the high usage rates of unsupported JDK versions imply that companies are getting support from third-party vendors like Amazon Corretto, Azul Zulu, and OpenLogic.

We expect adoption of Java 21 to increase as Oracle accelerates the frequency of long-term support JDK versions from every 3 years to every 2 years. That renewed commitment to long-term support may be a contributing factor to Oracle Java gaining more ground versus generic OpenJDK versions when compared to 2023 data. The burden of upgrading JDKs used to be very large due to large sets of features being released with each subsequent release. Today, many companies realize that it's much easier to maintain support for the latest version of Java due to a smaller group of features introduced with each release.

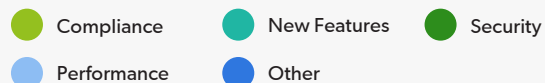
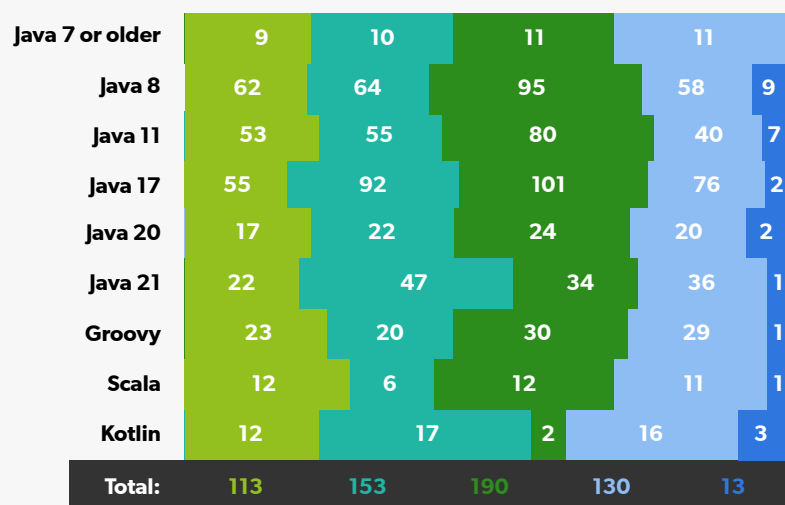
JDK Programming Language for Main Application



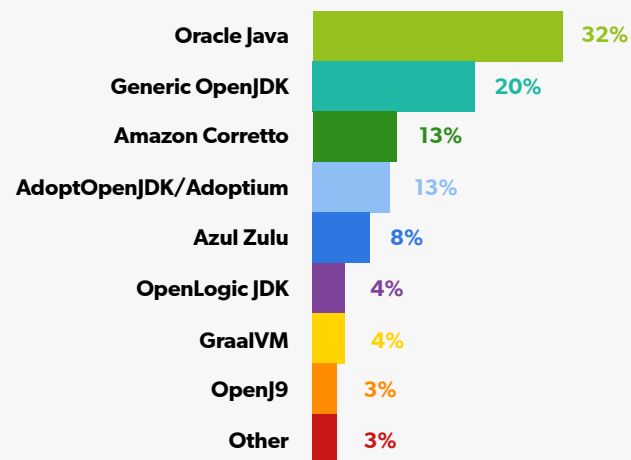
Of note were the cited reasons for upgrading and how they varied by JRE/JDK version. Overall, long-term support (25%) and security (24%) were top reasons for upgrading, but new features (19%), performance (16%), and compliance (14%) also made the list, with multiple responses possible.

The bottom line is that long-term support and security go hand in hand, and long-term support ensures patches and other enhancements to address novel security threats and protect your business application.

JDK Programming Language for Main Application



JRE/JDK Distribution



High usage rates of unsupported JDK versions convey that companies are getting support from third-party vendors like Amazon Corretto, Azul Zulu, and OpenLogic.

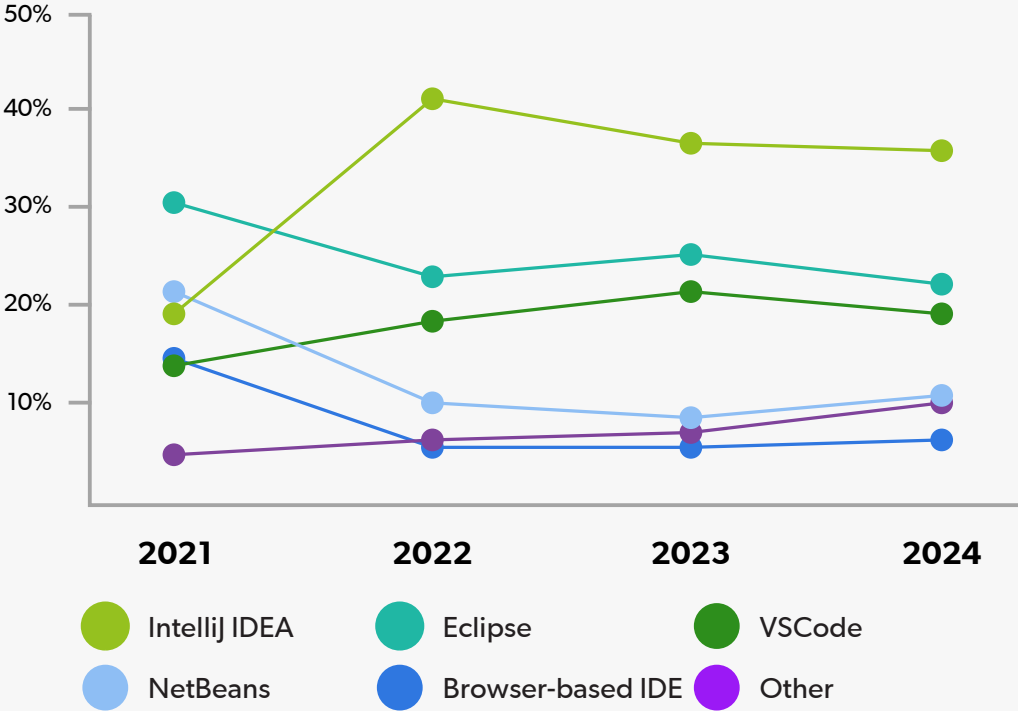
IDE INSIGHTS

Respondents were also asked to share which Java IDEs they were using. Once again, IntelliJ IDEA topped the charts with 41% of respondents. Eclipse held onto the second position (23%), but Microsoft Visual Studio Code, or VSCode, is close behind at 19%. Additionally, 84% of respondents using IntelliJ IDEA said they use more than one IDE in their Java development practice, with VSCode being the most popular secondary IDE.

VSCode integrates with other parts of the development toolchain, including multi-language environments and cloud-specific tooling. Microsoft released VSCode as a free solution in 2015 but didn't effectively reach the Java community until 2020. Fleet, an emergent code editor from JetBrains, is moving from beta to general availability later in 2024. With this development, the IDE breakdown may shift further in the coming year.

Fleet, an emergent code editor from JetBrains, is moving from beta to general availability later in 2024. With this development, the IDE breakdown may shift further in the coming year.

IDE Usage: 4-Year Comparison



APPLICATION SERVERS, FRAMEWORKS + OTHER TECHNOLOGIES

Outside of shifts in IDE preferences, much of respondents' Java tech stack remains status quo from prior years, with dominant technologies such as Tomcat, Spring Boot, and Jenkins overshadowing alternative options.

36% of respondents said they use Tomcat as the application server for their main application, with JBoss/Wildfly (15%), WebLogic (12%), WebSphere (10%), Jetty (10%), and Glassfish/Payara (8%) taking near even shares of the remainder. Another 7% respondents said they don't use an application server, and nearly a quarter of respondents selecting more than one application server option.

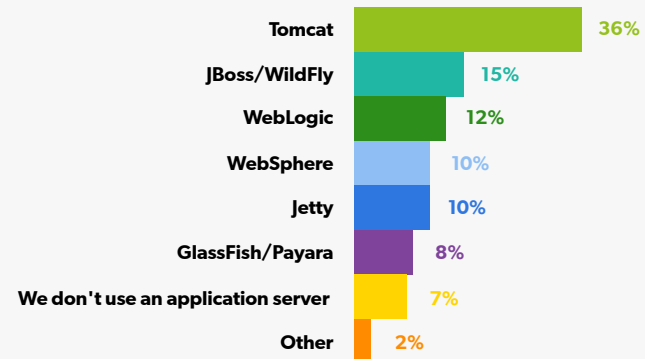
There are a variety of factors impacting these numbers. First, Tomcat is very robust with strong community support and regular updates. But more importantly, it's a free, mature, open-source enterprise solution, which sets it apart from JBoss, Weblogic, and Websphere. As development teams continue to transition to microservice based environments, this trend toward Tomcat will likely continue.

Results were similarly split for microservice frameworks, with 67% respondents using Spring Boot over alternative options such as DropWizard (11%), Quarkus (8%), Micronaut (5%), and Vert.x (1%).

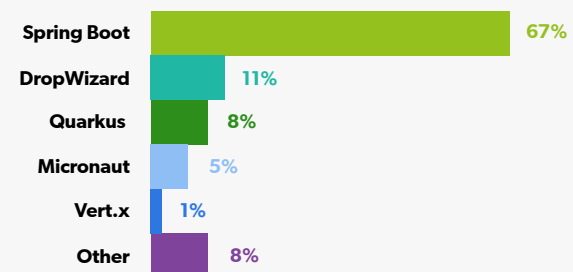
Jenkins was by far the most popular CI/CD technology at 37%. TeamCity saw usage more than double over 2023 (10%) while usage rates of other technologies remained largely stagnant. GitHub Actions (17%), Travis CI (9%), Circle CI (8%), and Bamboo (7%) rounded out secondary CI/CD options.

TeamCity saw usage more than double while other CI/CD technologies largely remained stagnant.

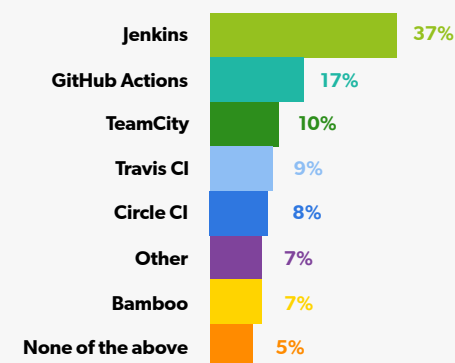
Application Server



Microservice Framework



CI/CD Technologies



Microservices + Cloud

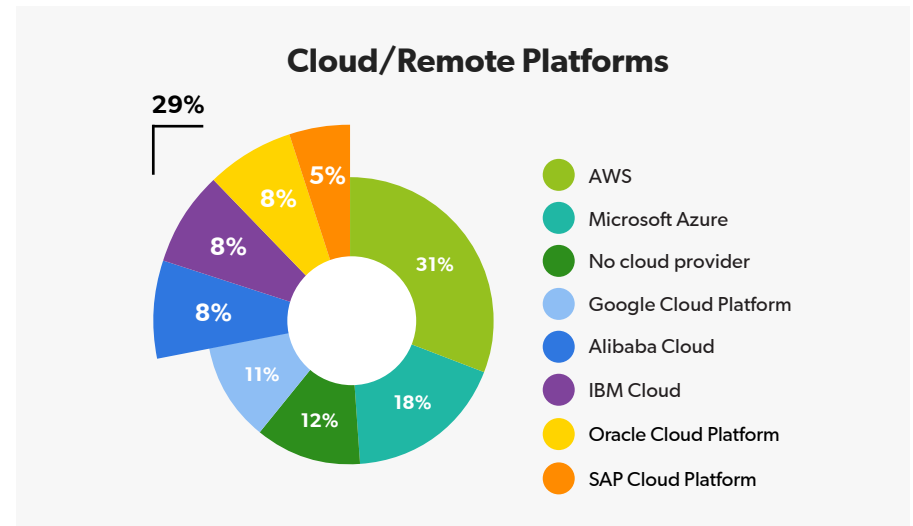
Microservices and cloud adoption rates often go hand in hand. This section dives into microservices adoption, cloud providers, average startup times for microservices applications, and redeploy times for containerized cloud environments.

MULTI-CLOUD ENVIRONMENTS

It comes as no surprise that AWS (Amazon Web Services) is by far the most popular cloud provider, representing 31% of respondents. But what does stand out is that the number of respondents who say they don't use any cloud providers has fallen to 13% versus 21% in 2023. Respondents could select more than one provider, with Microsoft Azure (18%) and Google Cloud Platform (11%) rounding out the "Big Three" of cloud providers.

More interesting than the leading cloud providers, however, is the popularity of secondary cloud platforms like Alibaba Cloud (8%), IBM Cloud (8%), Oracle Cloud Platform (7%), and SAP Cloud Platform (5%). Collectively, they rank higher than all but AWS and occupied specialized roles in companies' multi-cloud environments.

According to Gartner, more than 70% of enterprises will use industry cloud platforms to accelerate their business initiatives by 2027. These numbers will likely continue to grow as business needs increase.



MICROSERVICES TRENDS

After years of trending upward, microservice adoption rates were largely status quo. While more than half of respondents are on microservices, a sizeable portion are using monolith or modular monolith environments.

Survey responses show that most companies have either fully transitioned to microservices (42%) or are actively transitioning to microservices (45%). An additional 11% were currently in the planning stages, and 2% had reverted from microservices.

What was surprising is the number of microservices that respondents reported using in their applications, with 18% using 11-20 microservices and 23% using more than 21 microservices. Year-over-year data comparing the past 4 years reveals further insights, with the number of respondents using 11 or more microservices ballooning from 16% in 2021 to 41% in 2024.

Those numbers call into question whether those companies are truly implementing microservices, or whether those environments are more indicative of so-called miniservices. Microservices break down an application into small manageable chunks called services, whereas miniservices run portions of the application through a larger application server while keeping more regularly maintained aspects of the project into a microservice-like architecture. This data point also suggests microservice bloat as applications get older.

Enthusiasm for implementing microservices is often high, but companies find implementation difficult as they slide into miniservices or macroservices inadvertently—or make a conscious decision to revert from microservices entirely.

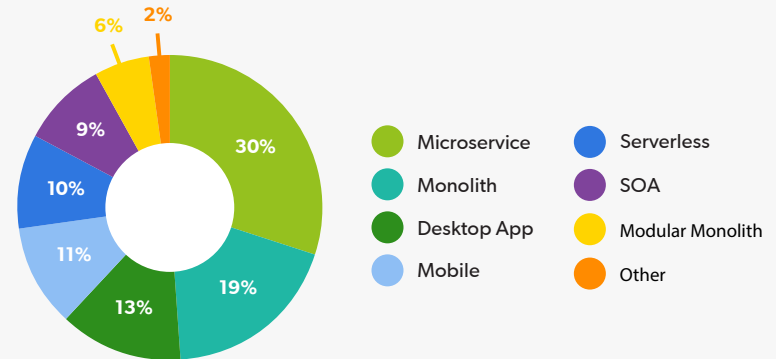
The number of respondents using 11 or more microservices has ballooned from 16% in 2021 to 41% in 2024.

Microservices: An architectural pattern where loosely coupled services communicate via lightweight protocols.

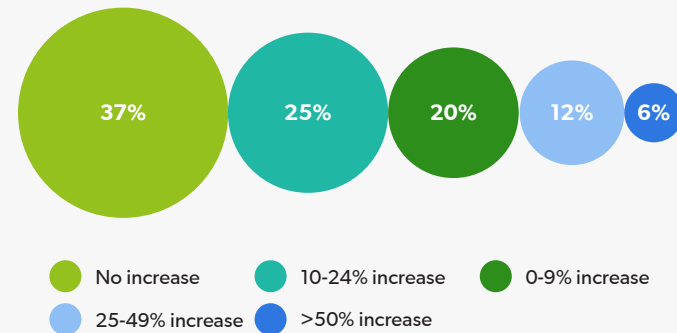
Miniservices: Also called modular monoliths, miniservices combine monolithic corporate or third-party services with more agile application architecture.

Macroservices: Monolithic applications that have been broken down into smaller monoliths.

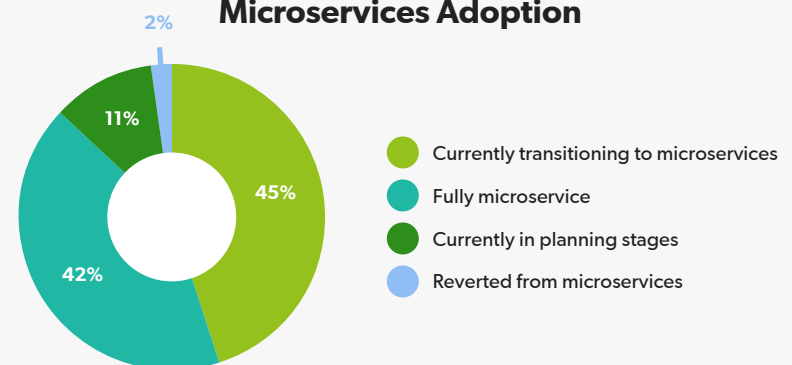
Application Architecture Type



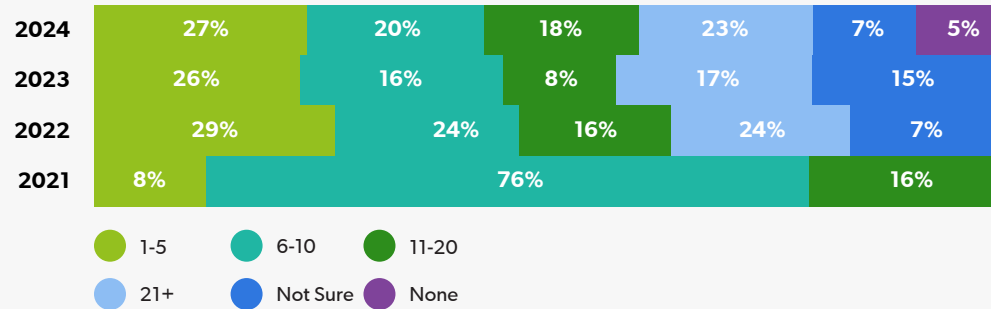
Increase in Startup Time for Microservice Applications



Microservices Adoption



Microservices

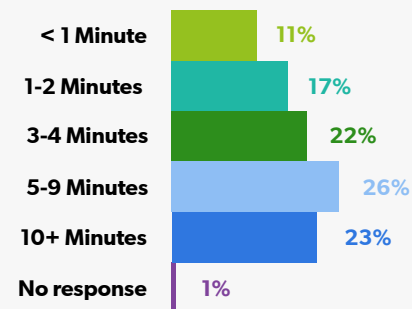


REMOTE DEPLOYMENT TRENDS

Microservices and containerized software trends go hand in hand. While 28% of remote containerized environments redeploy in under 2 minutes, 22% take 3-4 minutes, and nearly half (49%) take 5 minutes or more. Somewhat conversely, 57% of respondents said they hadn't seen an increase in the startup time for their microservice applications or that increase was by less than 10%, while almost half have seen startup times increase by 10% or more.

This could be due to a variety of factors. Consistent startup times could be due to a well-built team that's able to maintain a microservice application in the cloud. More likely, however, is that companies are continually incurring performance hits as they deploy and maintain applications in the cloud. This will make controlling the amount of resources sent to and from the cloud increasingly important.

Deployment Time for Remote Containerized Environments



Productivity Trends

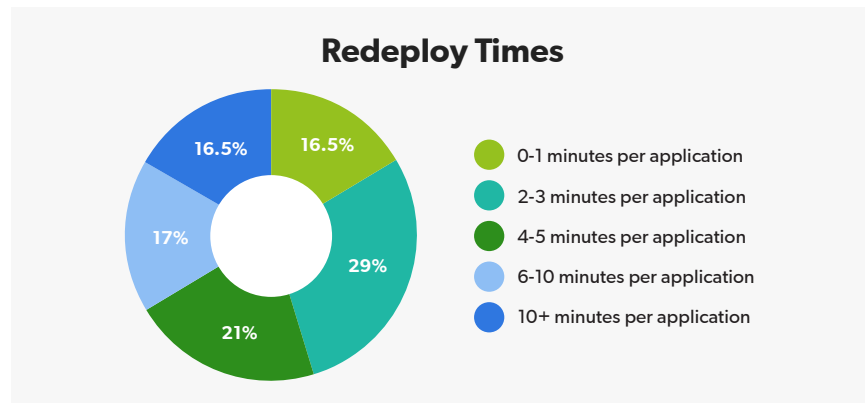
Java developers were asked what they would do with 10% more development time. Pragmatic answers like adding features (26%) and improving test coverage (18%) rose to the top, but write-in answers like “drink coffee” and “remove technical debt” won our hearts.

Read on for an in-depth look at the biggest time traps facing Java development teams and the tools and techniques they’re using to solve them.

REDEPLOY BENCHMARKS

Redeploys remain a threat to Java developer productivity, and each year, the Java Developer Productivity Report seeks to benchmark redeploy times. 17% reported redeploy times of less than 1 minute, 29% reported redeploy times between 2-3 minutes, 21% reported redeploy times of 4-5 minutes, 17% reported redeploy times of 6-10 minutes, and an additional 17% reported redeploy times lasting more than 10 minutes.

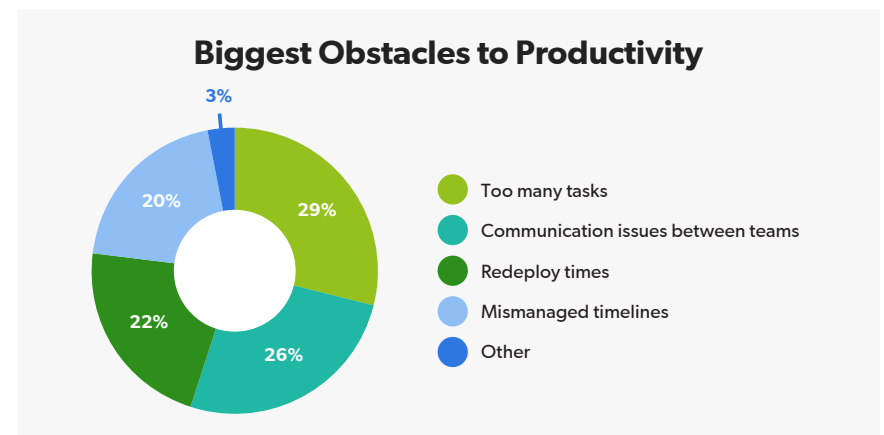
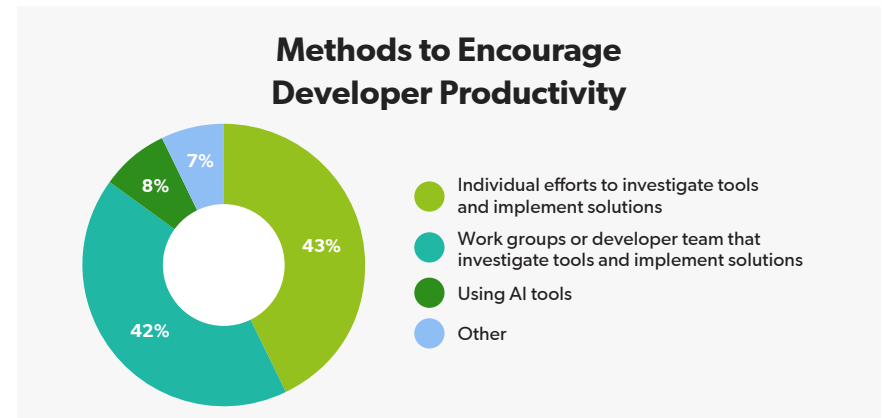
Those numbers may seem trivial in isolation, but can add up to a month or more of development time annually. Each redeploy not only represents a break in developer flow that makes your developers less productive, but also a crucial delay as businesses look to go to market faster. Increased productivity doesn’t just enable more time for innovation; it also improves developer job satisfaction.



AI IN JAVA DEVELOPMENT

Only a small subset of respondents (8%) reported using AI tools to assist in writing code, but those respondents were split between ChatGPT (and similar all-purpose generative AI tools) and specialized tools like GitHub CoPilot.

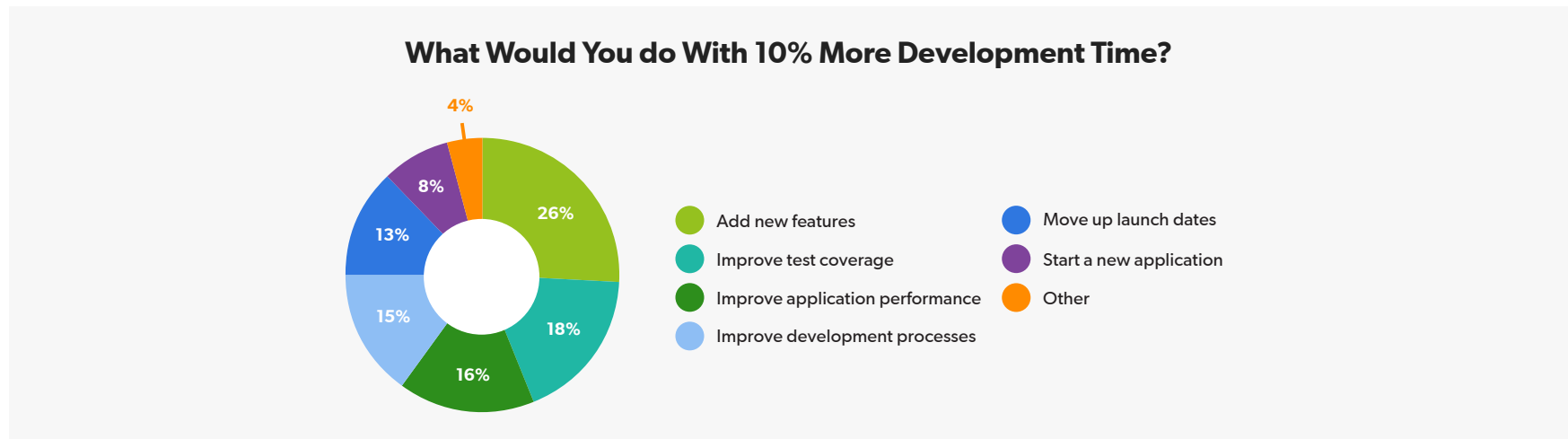
Some companies—especially in creative industries—remain skittish to incorporate AI tools out of concerns for plagiarism and data security, while others see generative AI as the next frontier to gain a competitive edge. It won’t be long until AI becomes table stakes and non-adopters will be left behind.



DEDICATED PRODUCTIVITY TEAMS

While they operate under a variety of names, including productivity teams and automation workgroups, they have a common focus: to investigate and implement new tools and workflows across the enterprise. At some enterprises, productivity workgroups are elective and a pathway for professional advancement, while others have dedicated productivity teams that report through engineering.

No matter the structure, we added this question to better understand the changing landscape on how development teams address productivity. This year's survey was the first time we asked this question and it was clear that productivity teams play a critical role in how companies solve their Java development challenges. 43% of respondents say that investigating and implementing tools is up to individual efforts, but nearly as many (42%) say their organizations use productivity teams to research solutions. That may be the relief developers need, as 29% said that too many tasks is their biggest barrier to Java development productivity.



Final Thoughts

Each year, the Java Developer Productivity Report produces powerful benchmarks on the challenges your Java development peers are facing—and the tools they're using to solve them.

While much of the Java development tech stack remains status quo, lightweight code editors like VSCode are gaining in popularity. Many companies have reached maturity with adopting microservices, but with a keen eye toward the number of microservices in an application. As cloud environments proliferate at the enterprise level, startup times for remote containerized environments are steadily increasing.

Redeploys remain a threat to Java developers' productivity—and cloud deployments are compounding that time drain with redeploy times exceeding 5 minutes. Companies are also taking new approaches to address productivity, including using AI to write code and forming productivity work groups to investigate and implement solutions on a broader level.

ABOUT JREBEL

JRebel is a Java developer productivity tool that allows developers to view code changes instantly and skip redeloys while maintaining application state. JRebel is trusted by leading brands worldwide to help Java developers write better code, faster.

Want to see how JRebel works on your project? Try it free for 14 days with no commitment.

TRY JREBEL FOR FREE

www.jrebel.com/free-trial