



# The Next Decade of Java

**By David Floyer**  
November 22, 2019

## PREMISE AND EXCERPT

The premise of this research is that Java is a leading application platform, has a good strategy for supporting the CI/CD model, and is embracing on-premises and cloud deployments. Wikibon believes that Java will be a strong foundation for all deployments and enterprise initiatives over the next decade.

The Wikibon financial analysis compares "upgrade-when-necessary" and "upgrade-every-release" Java strategies for a \$20 Billion enterprise. The study concludes an "upgrade-when-necessary strategy" is 31% less expensive, and has a \$40 million IT budget saving over four years.

As a result of the financial analysis, the Wikibon Java survey, and in-depth interviews with senior Java executives, Wikibon strongly recommends that most enterprise IT executives adopt an "upgrade-when-necessary" strategy for a mixed portfolio of Java applications as the most flexible and optimal strategy. Some enterprises may include keeping the upgrades always current for some part of the Java application portfolio. Wikibon strongly recommends the Java subscription services from Oracle or other vendors to support either of these strategies.

## THE JAVA PLATFORM

The Java platform is open-source software for developing applications and deploying them across different platform computing environments. The specifications of the [Java Community Process \(JCP\)](#) drive consistency across various ports and implementations.

Java is ubiquitous – all the way from enterprise mission-critical applications to IoT device management at the Edge. All the major cloud platforms, including AWS, Microsoft Azure, and Oracle Cloud use it extensively. The majority of enterprises use Java applications and Java desktop applications.

A recent Wikibon survey, focused on large enterprises with at least some Java applications, illustrates Java's importance. We asked participants to select the top two programming languages for their enterprise. The results in Figure 1 show that 83% of these respondents regard Java as one of their top two programming languages, significantly ahead of C++ (46%) and Python (32%).

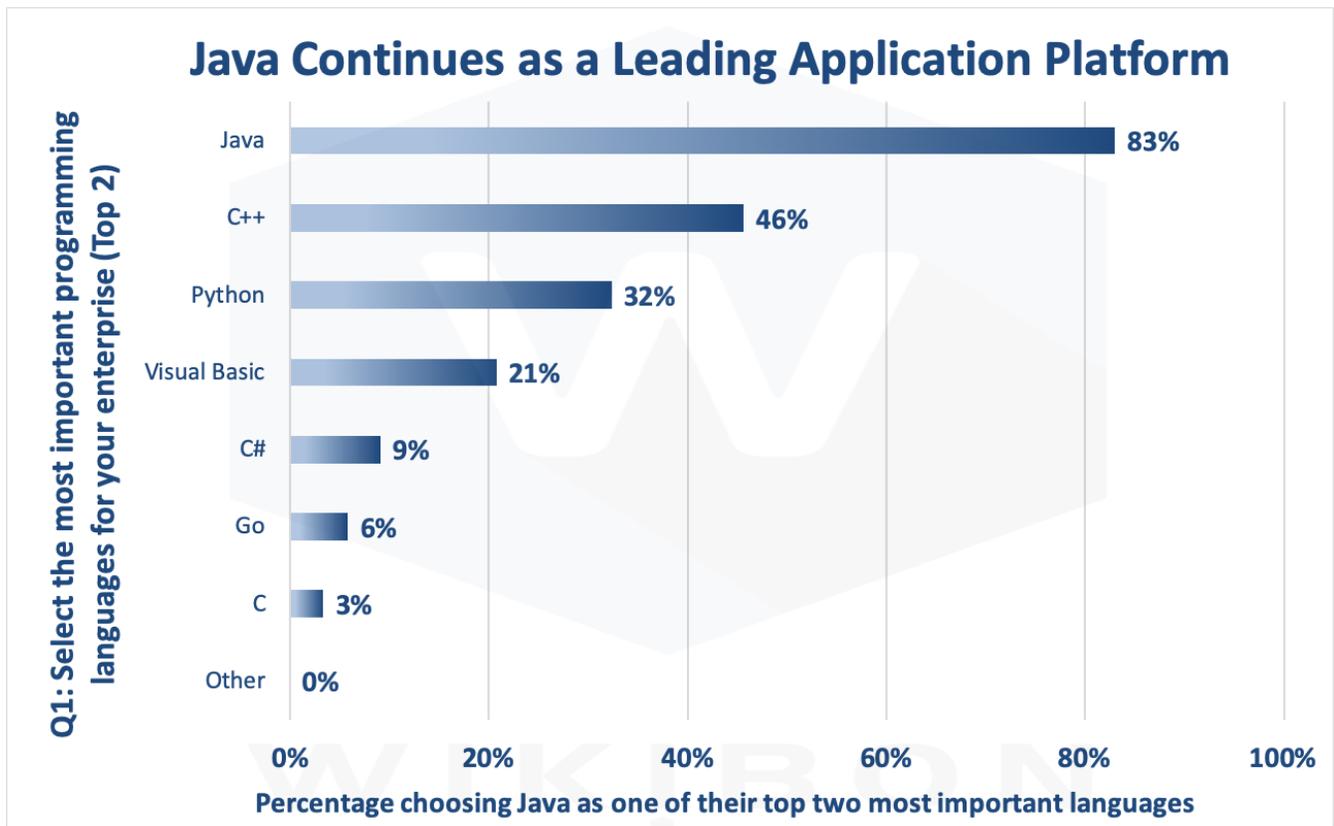


Figure 1 - Java Continues as a Leading Application Platform

Source: Wikibon © 2019. Survey of 122 large enterprises and 5 in-depth interviews, n=121.

The survey also confirmed that Java is important to developers and ISVs across all industries, company sizes, and government agencies (Figure 8 in Footnotes shows the details).

## JAVA AND CONTINUOUS DEVELOPMENT

The last decade has seen application development shift to mobile-first and browser-first. As a result of this and other factors, application development has moved to a Continuous Integration model with Continuous Delivery and/or Continuous Deployment (CI/CD, see Figure 2 below).

This requirement is putting stress on development organizations that rely on Java. The previous “major” release cycle cadence (every 2+ years) caused major disruptions in development, with all hands being dedicated to migration projects every few years.

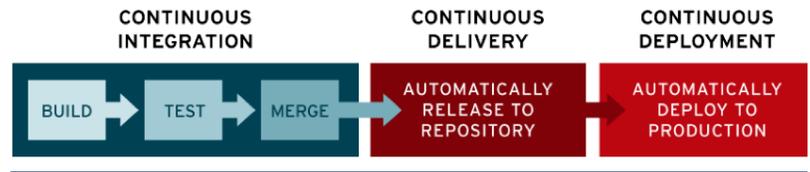


Figure 2 - Continuous Delivery and Deployment (CI/CD) Fundamentals.

Source: [Red Hat DevOps](#)

Development teams now strongly expect and prefer more frequent and manageable upgrades rather than infrequent-but-massive changes. The teams also need feature upgrades to be on a predictable timetable. As a result, the Java ecosystem has now opted for a six-month feature release cycle in support of a continuous development model.

Wikibon believes Java itself is now positioned to fully support modern CI/CD development models and to implement cloud native micro-services. It must remain relevant, performant, available, low-cost, and above all agile.

Wikibon observes that the Java ecosystem continues to be agile and will meet the future needs of developers, as the move to the new release cycle illustrates. Wikibon has confidence that Java will remain a leading development and deployment platform over the next decade.

## JAVA SHORTFALLS AND THE NEED FOR UPGRADES

In our survey we investigated the major reasons for Java upgrades. Figure 3 below shows that the leading application shortfalls are related to both application performance and security. Application performance issues include response time and stability. The security issues include ransomware, unauthorized intrusion into applications, and compromised customer data or IP loss. These security risks increase when systems and/or application code remain unpatched.

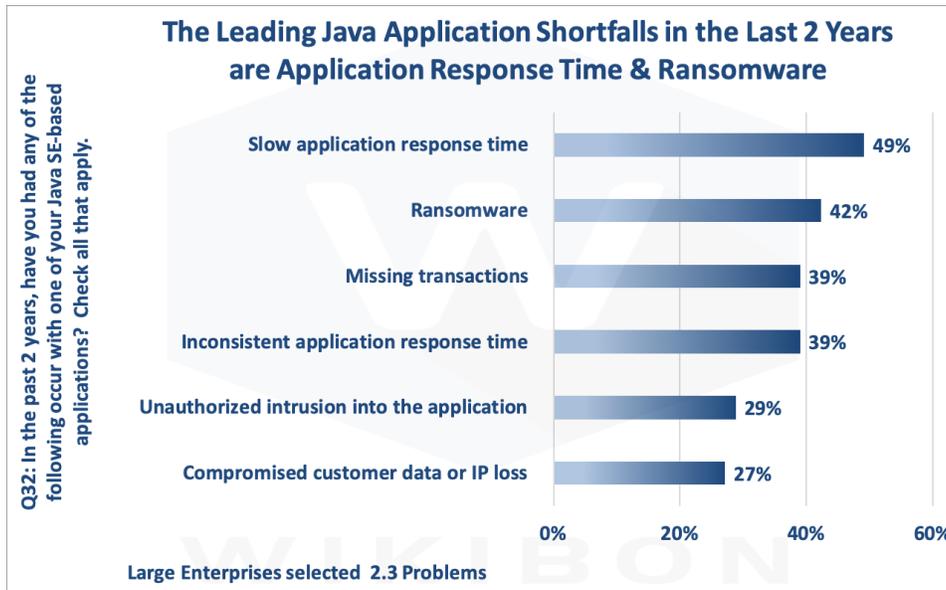


Figure 3 - Application Response Time & Ransomware are Leading Application Shortfalls

Source: Wikibon © 2019. Survey of 122 large enterprises and 5 In-depth Interviews. Selection is Employees ≥ 30,000, Java Applications ≥ 50, n=59.

## DRIVERS FOR JAVA UPGRADES

As such, Figure 4 below shows the most important reasons for actual Java upgrades are also related to application performance and security (the respondents can pick up to three reasons from the list). This is line with the shortfalls found in Figure 3.

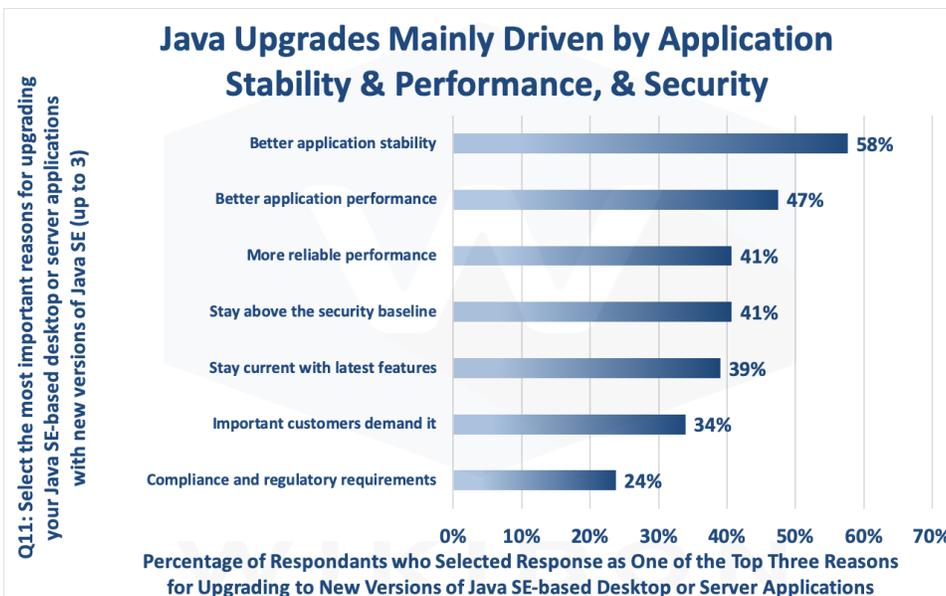


Figure 4 - Application Response Time & Security are Leading Drivers for Java Upgrades

Source: Wikibon © 2019. Survey of 122 large enterprises and 5 In-depth Interviews. Selection is Employees ≥ 30,000, Java Applications ≥ 50, n=59.

Figure 4 shows that better application stability, better application performance, and more reliable performance are the top three drivers of Java upgrades. Staying above the enterprise's security baseline is the next driver of importance.

Our research from in-depth interviews with senior Java executives in IT indicates upgrades are needed more frequently for business critical Java applications. In addition, the support for upgrades across the Java application portfolio needs to be more flexible.

This research supports the finding that stability, performance and security are paramount given that in business today there is low customer tolerance to IT failure in the face of increasing downtime and security threats. Java vendors must deploy support strategies that mitigate these business risks.

## JAVA SUPPORT REQUIREMENTS

### SUPPORT COSTS

Although the open-source software is free, support of open-source software in general and Java support in particular has never been without costs! Enterprises have always paid for third party support of open-source software. Red Hat for Linux is an obvious example. The same is true for Java, with support services from Azul, Oracle, Red Hat, Bellsoft and other vendors. Some vendors provide support via their other products, such as Red Hat through RHEL licensing.

### JAVA APPLICATION PORTFOLIO

In thinking about support, Wikibon believes that enterprises should consider the breadth of their entire Java application portfolio and the different requirements of each application. Some require mission-critical support services, where the latest release of Java could be very important, and external support services need to be available at short notice. At the other extreme, simple Java read-only applications may run infrequently and do not need attention except when they fail. Therefore, a workable support strategy needs a long-term model, which allows flexible support of a broad portfolio of Java application systems. Each application type in the portfolio will have different requirements and timescales for upgrades, updates and patches. Figure 5 below shows that enterprises use multiple approaches and support options for different parts of their Java portfolio.

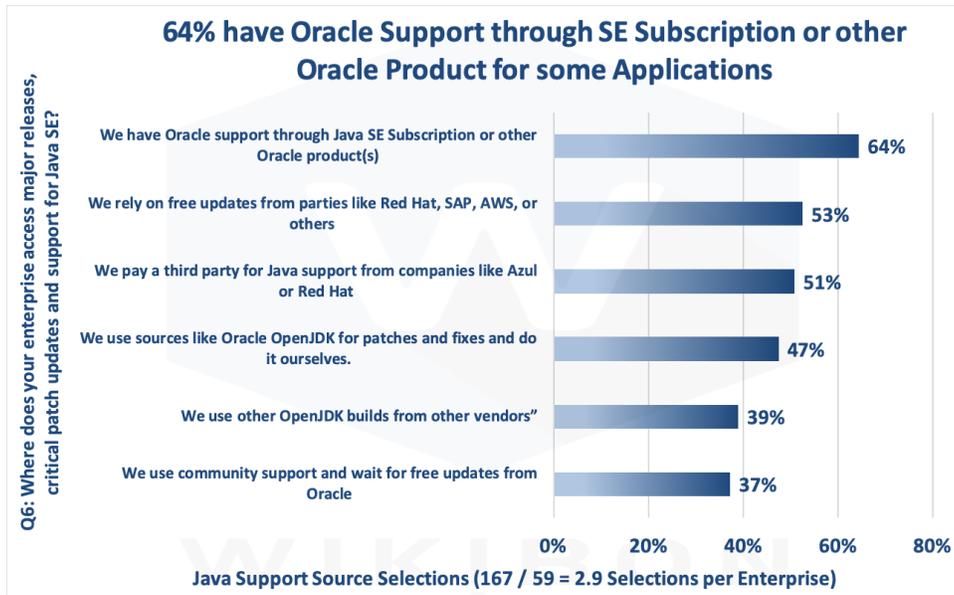


Figure 5 - Enterprises use Multiple Support Strategies for Different Parts of the Java Portfolio

Source: Wikibon © 2019. Survey of 122 large enterprises and 5 In-depth Interviews. Selection is Employees ≥ 30,000, Java Applications ≥ 50, n=59.

In Figure 6 below, respondents characterized the overall Java application portfolio upgrade strategy they employ to meet their overall application business objectives. It shows that application upgrade strategies need to be flexible to meet overall enterprise goals, and to enable different solutions for applications and application suites within a portfolio of Java systems.

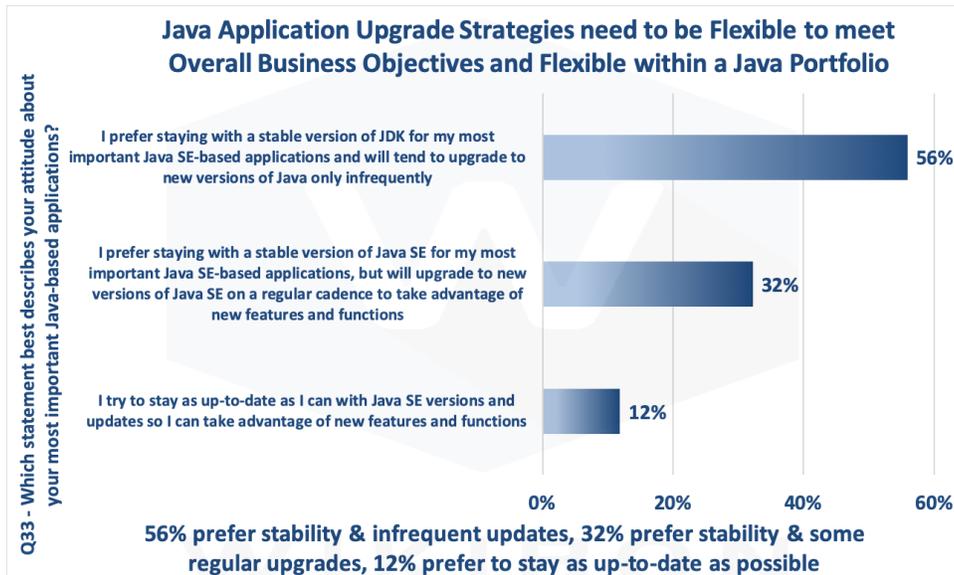


Figure 6 - Java Application Portfolio Upgrade Strategies

Source: Wikibon © 2019. Survey of 122 large enterprises and 5 In-depth Interviews. Selection is Employees ≥ 30,000, Java Applications ≥ 50, n=59.

## SO WHAT'S NEW? ORACLE JAVA SE SUBSCRIPTION

### NEW SUPPORT OFFERINGS

The recent change to a six-month Java release cycle and the “End of Public Updates” (EoPU) for Java 8 have radically changed the support requirements for users, resulting in new offerings from Java support vendors. To begin with, Oracle has retired its legacy and higher-cost perpetual licenses and separate support service. In its place, Oracle has launched the [Oracle Java SE Subscription](#), a simpler and relatively lower cost option, which includes the commercial license and support services as an annual subscription. This service is available via Oracle Store, Oracle Partner Network and Oracle direct sales.

Oracle has the advantage of being the primary steward of Java, and claiming it is likelier, with a larger installed base, to find problems before other providers and offer patches more rapidly. However, Wikibon expects other vendors to bring out their versions of a Java SE subscription product as well.

### STRATEGIC OPTIONS

Some enterprises might prefer to upgrade their Java portfolio on a six month upgrade cycle. This approach offers enterprises a predictable timeline, provides the opportunity to acquire new features earlier, and enables development teams to be more productive with modern application development capabilities. For some enterprises the costs of this strategy could look similar over a two to three year period compared to the costs of the one-time major upgrade cycles of the past.

The Java SE Subscription offers enterprises some important advantages over a six-month upgrade strategy. The subscription offering provides the flexibility to stay on older releases almost as long as required. It can also accommodate a mixed environment, which is likely the reality for most organizations with a large Java portfolio. The cost of this offering are predictable and for medium and large enterprises is likely to deliver overall savings.

In addition, our research shows that the Java SE Subscription is likely to be 31% less expensive than a six-month upgrade strategy, because of lower labor requirements. Beyond the cost savings of about \$40 Million over four years, a high level of support can provide best practice business protection against downtime, customer churn, reputation damage, and other business risks.

Whenever there is any change in product or support offerings, every enterprise, large and small, must consider how to address the opportunities and challenges of deciding how they will adapt to these changes. Alternatives should be evaluated in light of the strategic fit of the available options to their own business model, their own skill levels, and other factors. During our in-depth interviews, Wikibon found that the changes in the Java support options did not help every user – notably, some small ISV Java users.

However, for larger enterprises (the focus of this report), Wikibon believes there is almost always a clear and compelling case for adopting the cadence changes and the Java subscription option from Oracle or other vendors. This case includes enterprises that are taking conservative upgrade strategies, as well as enterprises taking aggressive upgrade strategies. In the next section, Wikibon examines the business case for the Oracle Java subscription option.

## FINANCIAL ANALYSIS OF JAVA SUPPORT STRATEGY OPTIONS

The four-year financial analysis in Figure 7 below represents the likely outcome for an enterprise with a revenue of \$20B, an IT budget of \$800M, and a total IT staff of 1,800. We estimated that about 450 of the IT staff would be supporting a Java portfolio of 521 applications. The source of all the detailed parameters in this model is the 2019 Wikibon survey of senior Java executives. The tables in the [Footnotes](#) below list the detailed assumptions and calculations.

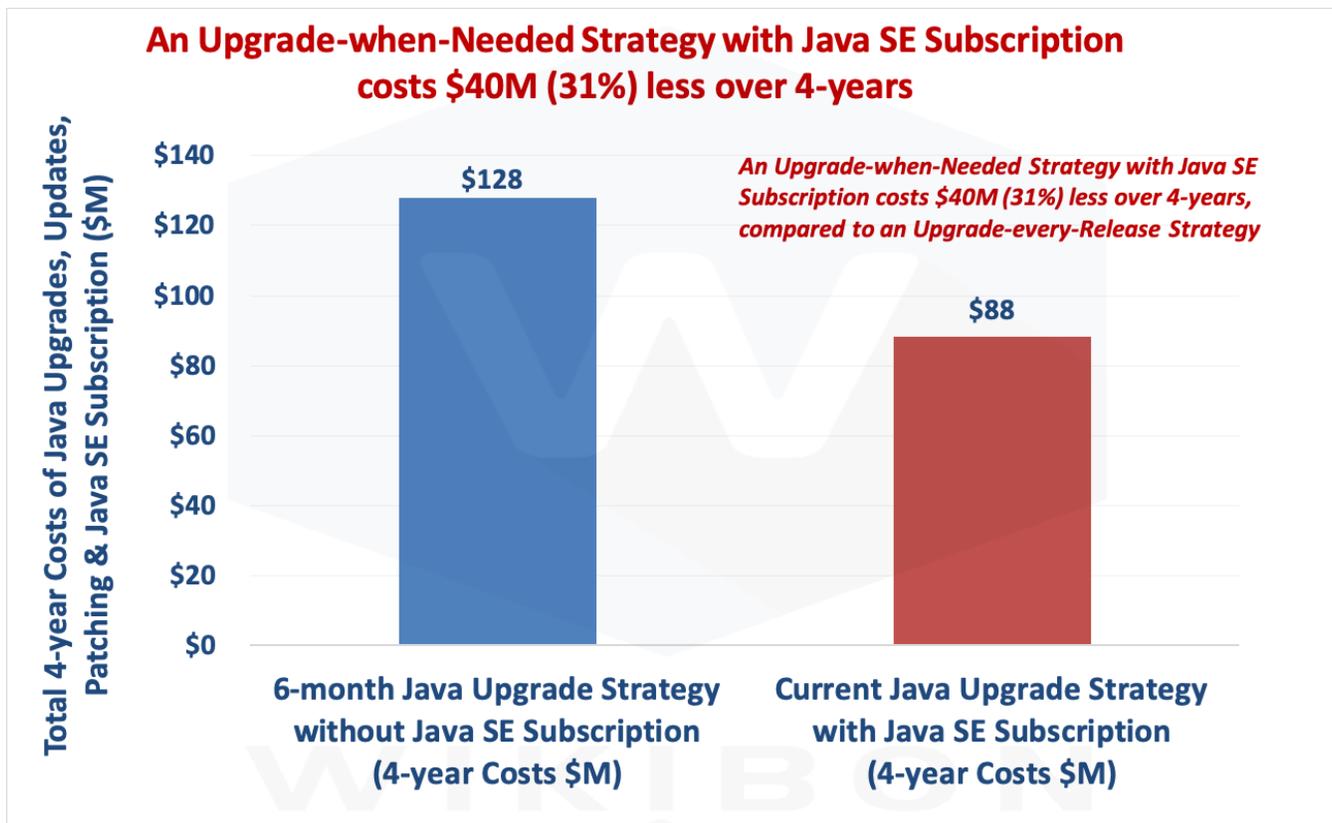


Figure 7 - Comparing a Java Upgrade-when-Needed Strategy with an Update-every-Release (Every Six-Months) Strategy

Source: Wikibon © 2019. The data comes from Table 4 in the [Footnotes](#) below.

The blue column in Figure 7 shows the costs of six-month Java upgrade strategy. This strategy upgrades all applications to the latest release as soon as it is available, and before support ends six months after release availability. The cost of each upgrade is lower than the previous two-plus year major upgrade strategy, and slightly less for updates and patches, but IT must perform more upgrades overall. Wikibon's analysis calculates the four year cost of this strategy to be \$128M.

The red column in Figure 7 shows lower costs (\$88M) for an upgrade-when-needed strategy supported by the Java SE subscription service. This approach allows IT to apply upgrades, updates, and patches selectively across their entire portfolio of Java applications. Some applications will need upgrading every six months, others less often, and some others may need upgrades only very rarely.

## FINANCIAL ANALYSIS CONCLUSIONS

Our conclusion is that the upgrade-when-needed approach will be \$40M less expensive over a four year period than upgrading every six months. It is important to emphasize that some enterprises will focus on reducing business risk as much as possible, and will be happy to spend the additional resources to maintain all or the majority of applications at the latest release level every six months. Even if this level of risk reduction is a priority, IT still benefit from the flexibility offered by a Java subscription service as well.

The tables in [Footnotes](#) below show the detailed calculations and assumptions used in the model. Figure 9 in the [Footnotes](#) shows additional line items of the data shown in Figure 7.

## CONCLUSION: JAVA WILL CONTINUE AS A LEADING APPLICATION PLATFORM OVER THE NEXT DECADE

Wikibon believes that the change to a six month release cadence is an essential change for Java to remain competitive and relevant compared to other CI/CD cloud application platforms. The advantages of this approach include establishing a predictable upgrade timeline as well as earlier access to new features. Without these changes, some enterprises will have to seriously consider converting to another development platform. As Wikibon has emphasized many times before, this a very expensive option, with significant business risk. Wikibon believes that the changes in Java support options will make the need for such a conversion highly unlikely.

The financial analysis illustrated in Figure 7 shows that costs are significantly lower (31%) if an enterprise avoids upgrading every six months and deploys the Java SE Subscription model. This is a more flexible and lower cost (\$40 Million) way of balancing the upgrade, update, and patching requirements within an application portfolio.

In summary, Wikibon strongly believes that Java will continue to be a leading application platform, and will provide good support for the strategic multi-cloud and hybrid-cloud corporate challenges ahead. Java will continue to support enterprise requirements over the next decade, and its continued popularity is likely to ensure good and varied vendor support services.

## ACTION ITEM

As a result of our survey and financial analysis, Wikibon strongly recommends that most enterprise IT executives adopt an “upgrade-when-necessary” strategy for a mixed portfolio of Java applications as the most flexible and optimal approach. Other enterprises may elect to upgrade some part of the their Java application portfolio every six months. Wikibon strongly recommends the Java SE subscription services from Oracle or other vendors to support either of these strategies.

FOOTNOTES

JAVA INDUSTRY DISTRIBUTION

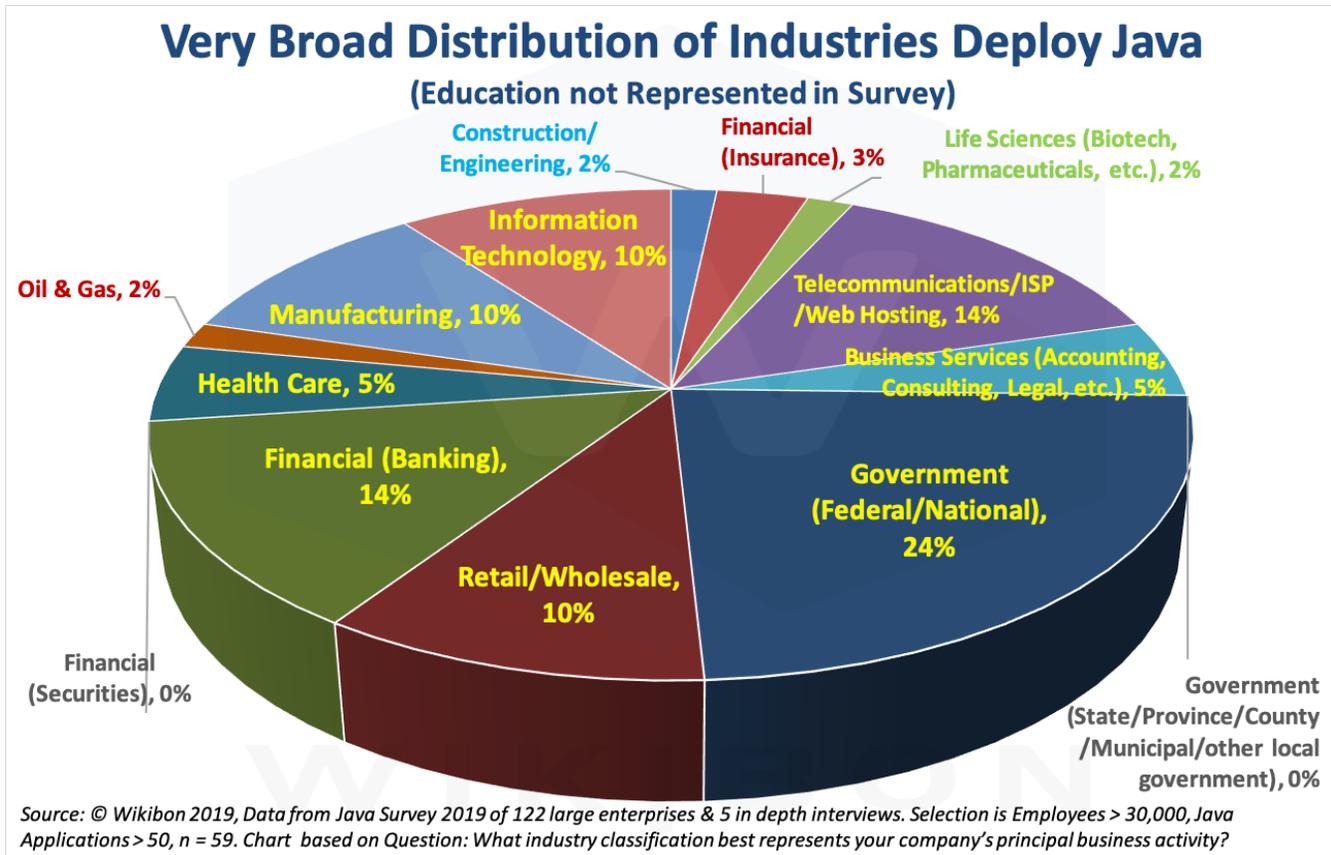


Figure 8 - Very Broad Distribution of Industries Deploy Java

Source: Wikibon © 2019. Survey of 122 large enterprises and 5 in-depth interviews, n=121.

DETAILED CALCULATIONS AND ASSUMPTIONS USED IN THE FINANCIAL MODEL

The charts and tables below are the source of the Financial Analysis of Java Support Strategy Options section and Figure 7 above.

Sources	Current Stable Large Enterprise Java Environment using Java SE Support - Base Business Metrics & Java Platform Resources	
Survey with employees ≥ 30,000, Java Apps ≥ 50	Number of Employees (or Equivalent)	62,882
Wikibon CIO Surveys	Revenue per Employee	\$318,000
Calc	Total Revenue (\$Billion)	\$20.0
4% Revenue - Rich IT Environment	IT Budget (\$Million) - Higher level of IT investment at 4% of Revenue	\$800
Salary \$100K, 50% overhead, 35% IT Budget	Internal IT Staff	1,866
	<b>Number of Java Apps in Portfolio</b>	
Survey Q3	Java 8 or earlier	168
Survey Q4	Java 11 (LTS release)	165
Survey Q5	Java 9, 10, 11, 12+ (non LTS releases)	188
Survey Q2	Total Number of Java Apps	521
Survey Q7&8	Internal Staff focused on Java Platform, Patches, Updates, & Upgrades	249
Survey Q7&8	Internal Staff focused on New Java App Development & Java App Extensions	203
Survey Q8	Total Number of Internal Java Staff	452
Calc	Other Internal IT Staff	1,415
Survey Q12&13	Internal Resources for Java Upgrade for 15% Most Important Java Applications (Person Years)	38
Wikibon Estimate 1/2 resources for 85% = 2.8x	Internal Resources for Java Upgrade for 85% Less Important Java Applications (Person Years)	108
Survey Q15&16	External Resources for Java Upgrade for 15% Most Important (Person Years)	30
Wikibon Estimate 1/2 resources for 85% = 2.8x	External Resources for Java Upgrade for 85% Less Important Java Application (Person Years)	84
Survey Q24	Internal Resources for Updating & Patching Java Applications (Person Years/ Year)	103
Survey Q27	External Resources for Updating & Patching Java Applications (Person Years/ Year)	20
Survey Q9 + Q10	Time between Most Important Java Application Upgrades (Months)	33
Survey Q9 & Q10 & Wikibon Analysis	Time between Less Important Java Application Upgrades (Months)	40
Survey Q17	Elapsed Time to Complete Java Application Upgrade (Months)	6.2

Table 1 - Wikibon Java Survey Results and Calculations

Source: Wikibon © 2019. Survey of 122 large enterprises and 5 In-depth Interviews.  
Selection is Employees ≥ 30,000, Java Applications ≥ 50, n=59.

Source	Current Stable Large Enterprise Java Environment using Java SE Support	
	<b>Cost of Java Version Upgrade for Most Important Applications (15% of 521 Java Apps = 78)</b>	
a	Number of Internal Person Years for Upgrading Most Important Applications	38.1
b	Cost of Upgrade Persons (including 50% overhead)	\$135,000
c	Number of External Person Years for Upgrading Most Important Applications (Consultant)	8.9
d	Cost of External Specialist Consultant Upgrade Persons	\$200,000
e	Number of External Person Years for Upgrading Most Important Applications (Outsourced)	20.7
f	Cost of External Outsourced Upgrade Persons	\$33,750
g = (a x b) + (c x d) + (e x f)	<b>Total Cost of Upgrade of Most Important Applications</b>	<b>7,613,390</b>
h	Time between Upgrades (Months)	33
j = g x 12 / h	<b>Total Cost of Java Upgrades for Most Important Applications/ Year</b>	<b>\$2,765,665</b>
	<b>Cost of Java Version Upgrade for Less Important Applications (85% of 521 Java Apps = 443)</b>	
k	Number of Internal Person Years for Upgrading Less Important Applications	107.9
b	Cost of Upgrade Persons (including 50% uplift for overheads)	\$135,000
m	Number of External Person Years for Upgrading Less Important Applications (Consultant)	25.1
d	Cost of External Specialist Consultant Upgrade Persons	\$200,000
n	Number of External Person Years for Upgrading Less Important Applications (Outsourced)	58.6
f	Cost of External Outsourced Upgrade Persons	\$33,750
p = (k x b) + (m x d) + (n x f)	<b>Total Cost of Upgrade of Less Important Applications</b>	<b>\$7,003,375</b>
q	Time between Upgrades (Months)	40
r = p x 12 / q	<b>Total Cost of Java Upgrades for Less Important Applications/ Year</b>	<b>\$2,101,012</b>
s	Number of Internal Person Years for Updates & Patching/ Year	103
b	Cost of Internal Staff for Updates & Patching / Year	\$135,000
t	Number of External Person Years for Updates & Patching/ Year	20
f	Cost of Outsourced External Staff for Updates & Patching / Year	\$33,750
u = (s x b) + (t x f)	<b>Total Cost Java Updates &amp; Patching/ year (External &amp; Internal)</b>	<b>\$14,568,230</b>
= a + k + s	<b>Number of Internal Persons Years for Upgrades, Updates, &amp; Patching/ Year</b>	<b>249</b>
= c + e + m + n + t	<b>Number of External Person Years for Updates &amp; Patching/ Year</b>	<b>133</b>
= j + r + u	<b>Total Cost of Maintaining Java Performance, Functionality, Security, &amp; Compliance/ Year</b>	<b>\$19,434,907</b>
	<b>Percentage of Internal Java Staff Dedicated to Java Upgrades, Updates &amp; Patching</b>	<b>55%</b>
	<b>Percentage of Total IT Budget</b>	<b>2.4%</b>

Table 2 - Calculations of Line items in Current Stable Large Enterprise Java Environment using Java SE Support

Source: Wikibon © 2019. Survey of 122 large enterprises and 5 In-depth Interviews.  
Selection is Employees ≥ 30,000, Java Applications ≥ 50, n=59.



Number of Internal Persons Years for Upgrades, Updates, & Patching/ Year		
	<b>Impact on Large Enterprise Java Environment Implementing an Upgrade Every Six Months Strategy</b>	
	<b>Cost of Java Version Upgrade for Most Important Applications (15% of 521 Java Apps = 78)</b>	
v (IDI & Wikibon Estimate)	Reduction Factor when moving Most Important Applications to Upgrade every 6 Months	1.5
= a / v	Number of Internal Person Years for Upgrading Most Important Applications / Reduction Factor	25.4
b	Cost of Upgrade Persons (including 50% overhead)	\$135,000
= c / v	Number of External Person Years for Upgrading Most Important Applications ( Consultant) / Reduction Factor	5.9
d	Cost of External Specialist Consultant Upgrade Persons	\$200,000
= e / v	Number of External Person Years for Upgrading Most Important Applications (Outsourced)	13.8
f	Cost of External Outsourced Upgrade Persons	\$33,750
as above	<b>Total Cost of Upgrade of Most Important Applications</b>	<b>\$5,075,594</b>
as above	<b>Time between Upgrades (Months), determined by when support ends</b>	<b>6</b>
as above	<b>Total Cost of Java Upgrades for Most Important Applications/ Year</b>	<b>\$9,831,793</b>
	<b>Cost of Java Version Upgrade for Less Important Applications (85% of 521 Java Apps = 443)</b>	
v (IDI & Wikibon Estimate)	Reduction Factor when moving Less Important Applications to Upgrade every 6 Months	1.5
= k / v	Number of Internal Person Years for Upgrading Less Important Applications / Reduction Factor	71.9
b	Cost of Upgrade Persons (including 50% uplift for overheads)	\$135,000
= m / v	Number of External Person Years for Upgrading Less Important Applications (Consultant) /Reduction Factor	16.7
d	Cost of External Specialist Consultant Upgrade Persons	\$200,000
= n / v	Number of External Person Years for Upgrading Less Important Applications (Outsourced) / Reduction Factor	39.1
f	Cost of External Outsourced Upgrade Persons	\$33,750
as above	<b>Total Cost of Upgrade of Less Important Applications</b>	<b>\$4,668,916</b>
as above	<b>Time between Upgrades (Months), determined by when support ends</b>	<b>6</b>
as above	<b>Total Cost of Java Upgrades for Less Important Applications/ Year</b>	<b>\$9,044,030</b>
w	Saving in Updates & Patching after new updates	10%
= s x (1 - w)	Number of Internal Person Years for Updates & Patching/ Year	93
b	Cost of Internal Staff for Updates & Patching / Year	\$135,000
= t x (1 - w)	Number of External Person Years for Updates & Patching/ Year	18
f	Cost of Outsourced External Staff for Updates & Patching / Year	\$33,750
as above	<b>Total Cost Java Updates &amp; Patching/ year (External &amp; Internal)</b>	<b>\$13,111,407</b>
as above	<b>Number of Internal Persons Years for Upgrades, Updates, &amp; Patching/ Year</b>	<b>281</b>
as above	<b>Number of External Person Years for Updates &amp; Patching/ Year</b>	<b>77</b>
as above	<b>Total Cost of Maintaining Java Performance, Functionality, Security, &amp; Compliance/ Year</b>	<b>\$31,987,229</b>
as above	<b>Percentage of Internal Java Staff Dedicated to Java Upgrades, Updates &amp; Patching</b>	<b>62%</b>
	<b>Percentage of Total IT Budget</b>	<b>4.0%</b>

Table 3 – Summary of IT Budget Savings and additional Business Value from IT after Migrating to a Cloud-first On-premises Hybrid Cloud.

Source: Wikibon © 2019. Survey of 122 large enterprises and 5 In-depth Interviews. Selection is Employees ≥ 30,000, Java Applications ≥ 50, n=59.

CONTINUED ON NEXT PAGE.



Comparative Four-year Cost of Java Upgrades, Updates & Patching	
<b>Current Java Upgrade Strategy with Java SE Subscription</b>	<b>4-year Costs</b>
Cost of Most Important Application Java Updates	\$11,062,658
Cost of Less Important Application Java Updates	\$8,404,050
Cost Java Updates & Patching/ year (External & Internal)	\$58,272,920
Cost of Oracle Java SE Subscription Support (or Equivalent Enterprise Agreement)	\$10,557,000
<b>Total Cost of Java Upgrades, Updates, &amp; Patching with Current Upgrade Strategy</b>	<b>\$88,296,628</b>
<b>6-month Java Upgrade Strategy without Java SE Subscription</b>	<b>4-year Costs</b>
Cost of Most Important Application Java Updates	\$39,327,171
Cost of Less Important Application Java Updates	\$36,176,118
Total Cost Java Updates & Patching/ year (External & Internal)	\$52,445,628
Cost of Oracle Java SE Subscription Support (or Equivalent Enterprise Agreement)	\$0
<b>Total Cost of Java Upgrades, Updates, &amp; Patching with 6-month Upgrade Strategy</b>	<b>\$127,948,917</b>
<b>Total 4-year Savings from Use of Java SE Subscription Support</b>	<b>\$39,652,289</b>

Table 4 – Summary of IT Budget Savings and additional Business Value from IT after Migrating to a Cloud-first On-premises Hybrid Cloud.

Source: Wikibon © 2019. Table 2 and Table 3 in Footnotes above are the data Sources.

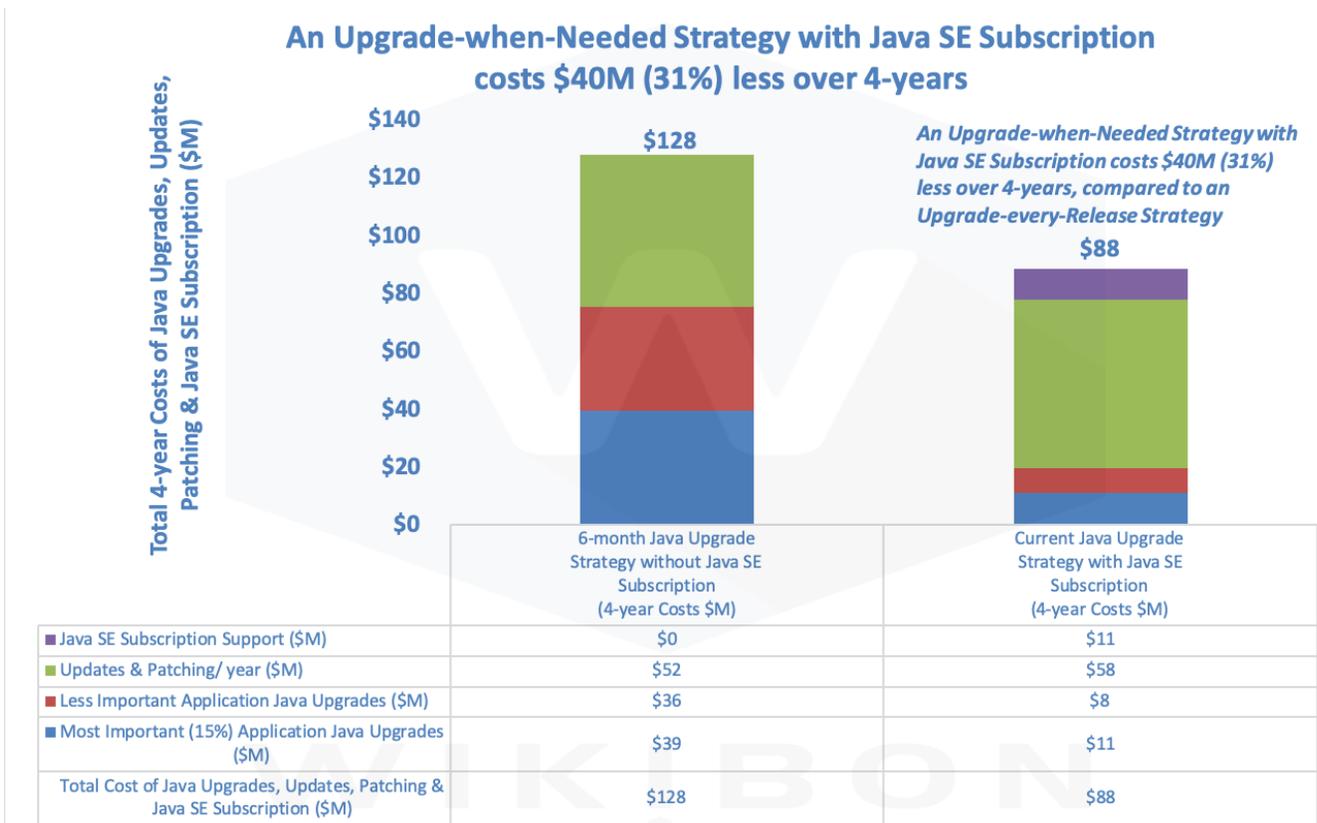


Figure 9 - Comparing Upgrade-when-Needed vs. Upgrade-every-Release (every Six months) Java Strategies, with Additional Line Items.

Source: Wikibon © 2019. Table 4 in the Footnotes above is the data source.

## WIKIBON TEAM



### David Floyer

**Chief Technology Officer**

@dfloyer

david.floyer@wikibon.org

David Floyer spent more than 20 years at IBM, holding positions in research, sales, marketing, systems analysis and running IT operations for IBM France. He worked directly with IBM's largest European customers, including BMW, Credit Suisse, Deutsche Bank and Lloyd's Bank. Floyer was a Research Vice President at International Data Corporation (IDC) and is a recognized expert in IT strategy, economic value justification, systems architecture, performance, clustering and systems software.