Java for Mid-sized Enterprises: On-premises & in the Cloud

By David Floyer

August 23, 2020

PREMISE

This report is an update to earlier Wikibon research, "The Next Decade of Java", which focused on Java for Global 2000 enterprises. This research report is in two parts.

Part 1 focuses on Java SE in mid-sized enterprises with less than 10,000 employees. The premise is that the 2018 change to a continuous improvement strategy for Java with releases every six months, together with the Java SE subscription services, will reduce costs for mid-sized enterprises.

Part 2 focuses on Java SE futures on-premises and in the cloud. Most enterprises want the flexibility to move workloads to the cloud, including Java workloads. The premise is that the change to a six-month Java release cadence will enhance current and future Java functionality for mid-sized enterprises. In particular, future releases will enhance Java to be even more efficient and scalable for both private clouds, public clouds, and personal devices such as laptops and computers. In addition, the inherent portability of Java will ensure that any migration costs between private and public clouds are minimized.

The overall Wikibon premise is that Java will continue as a strong foundation for high-security and high-performance applications for on-premises and cloud deployments for all enterprise sizes over the next decade.

EXECUTIVE SUMMARY

JAVA SE SUPPORT STRATEGY BUSINESS CASE

Figure 1 shows that Oracle's new Java SE Subscription offering can create value for mid-size enterprises. The vertical (y) axis shows the total 4-year costs of Java Upgrades, Updates, Patching, and a Java SE Subscription for a mid-sized enterprise. This assumes an annual enterprise revenue of $2.5 billion with 8,300 employees supporting 272 Java applications. Table 1 in the Footnotes section below gives the details of all the assumptions and how the data was derived from the Wikibon Java survey and 10 in-depth interviews.

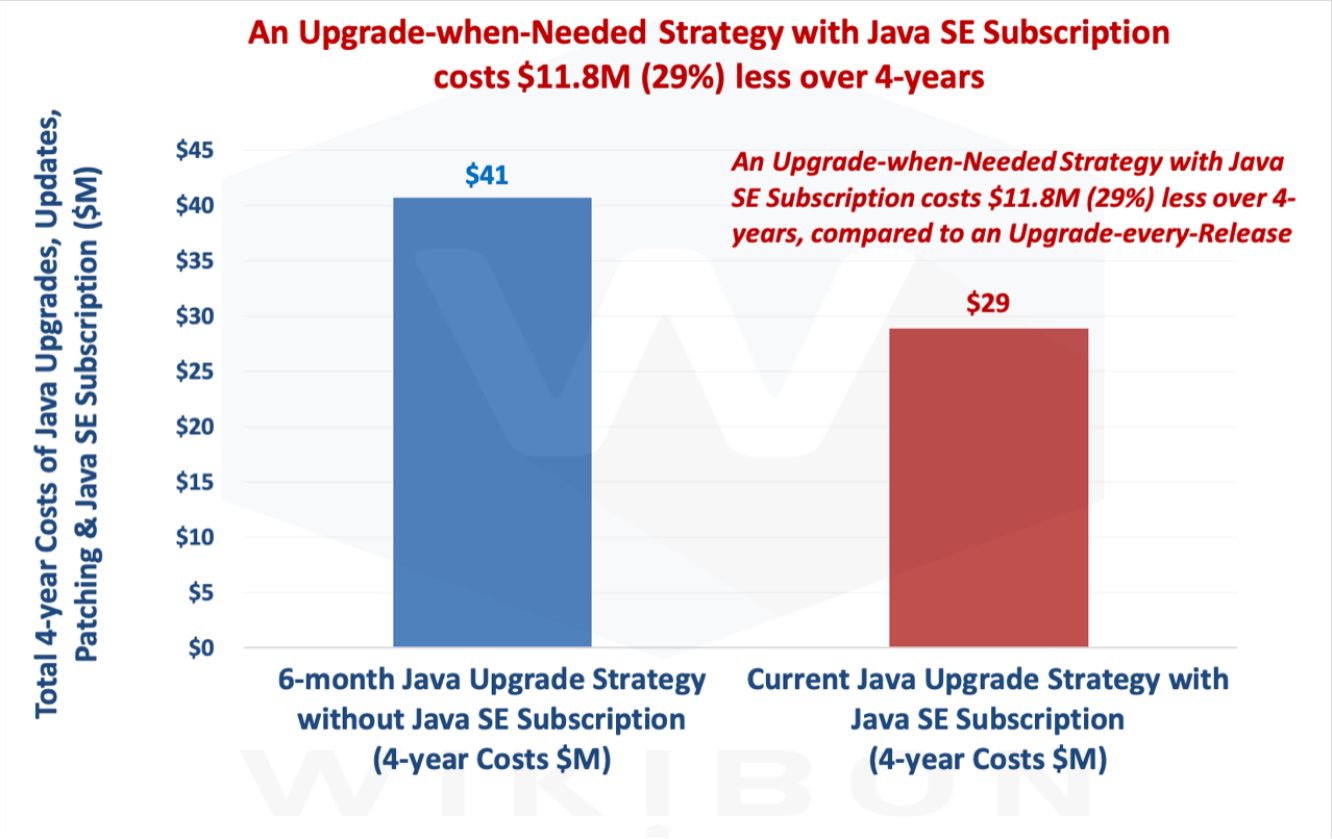

Figure 1 - Comparing Business Case for "Upgrade when Needed" vs. "Upgrade every Release" (Every Six-Months) Java Strategies

Source: Wikibon © 2020. Survey of 122 enterprises with more than 1,000 employees in 2019 and 10 in-depth interviews in 2019 and 2020. Selection is Employees ≥1,500 ≤10,000, Java Applications ≥ 18, n=45. The data comes from Table1 though Table 4 in the Footnotes below.

The x-axis shows two (2) scenarios. The left-hand scenario in blue is the total 4-year IT cost of $41 million for a Java support strategy upgrading to the latest release every six months. The right-hand scenario in red is the total 4-year IT cost of $29 million for a Java support strategy using Java SE Subscription. The Subscription enables support to be available for all releases for longer periods and specific releases for extended periods. The business case shows a saving of $11.8 million over four (4) years (a 29% saving) with the Java SE Subscription compared to an alternative strategy of upgrading every six months to stay current.

There may be some mid-sized enterprises where the benefit of upgrading to the latest release will be worth the additional cost; Wikibon's view is that most enterprises will be better served with the more flexible Java SE Subscription offering.

JAVA FUTURES

Oracle's roadmap for Java SE rolls out future Java enhancements that will continue to support an evolving application development environment, which, in turn, will support both current on-premises and future Java cloud-native applications.

Wikibon concludes that the 2018 change to a continuous improvement strategy for Java, together with the Java SE Subscription offering, is working for mid-sized enterprises. Wikibon projects that Java will continue to be a platform of choice as workloads move to the cloud. Wikibon also concludes that Oracle's more flexible upgrade strategy for Java introduced two years ago will lead to accelerated functional delivery.

Wikibon believes that Java will continue as a strong foundation for high-security and high-performance applications deployed on-premises and in the cloud for all enterprise sizes over the next decade.

RESEARCH METHODOLOGY

MID-SIZE ENTERPRISE SURVEY EXTRACT

For this research, we have defined a "typical" mid-sized enterprise as having revenues in the order of $2.5B and employing about 8,000 persons (revenue per employee is about $300,000). The annual IT budget is assumed to be $87 million (about 3.5% of revenue). The internal IT staff is assumed as 286 persons. The full details and calculations are shown in Table 1 in the Footnotes section below. The major data source used in this research was a Wikibon survey of 122 enterprises with more than 1,000 employees using Java in 2019. The survey was web-based, with four (4) introductory questions and thirty-three (33) detailed questions. All the 122 survey participants answered all of the total of thirty-seven (37) questions.

For this research report, we selected the responses of mid-range enterprises with between 1,500 and 10,000 employees, with at least eighteen (18) Java applications. The number of enterprises that met these criteria was forty-five (45). A small number of survey results were not included if the answers to questions were well outside normal expectations.

IN-DEPTH INTERVIEWS

Also, Wikibon conducted ten in-depth interviews with senior executives responsible for Java in 2019 and 2020. These participants were from different enterprises than those included in the survey. The interviews were about an hour-long and allowed the research team to develop and test different hypotheses based on the survey results.

An outside company arranged for the interviewees, who were guaranteed anonymity. The interviewees received a small consideration for sharing their expertise and time.

FINANCIAL MODEL

Wikibon developed the financial model for this report based mainly on the answers to the survey questions. Wikibon's ten years of experience building financial models of IT organizations were used to augment the model.

Our survey indicated that for a mid-sized enterprise of the size defined above, about 171 of the IT staff were supporting a Java portfolio of 272 applications. The survey shows that eighty-three (83) applications are on Java 8 or earlier, 84 on LTS release Java 11, and 105 on later Java versions. The full details and assumptions are shown in Table 2 in the Footnotes section below.

RESULTS OF SELECTED SURVEY QUESTIONS FOR MID-SIZED ENTERPRISES

Wikibon selected four (4) questions that illustrated important aspects of the survey. The following Figures 2 through Figure 5 show the question, the choices for selection, and the percentage distribution of the responses for mid-sized enterprises.

The previous research shows the same questions as answered by the Global Fortune 2000 enterprises. We've noted where there are significant differences between the two enterprise-sized groups.
Importance of Java to Mid-sized Enterprises.

Figure 2 below focuses on the answers to Question 1 (Q1).

SEE FIGURE 2 ON NEXT PAGE

• Q1: "What are the most important programming languages for your enterprise? Please select the top 2."
• The choices are Java, Python, C, C++, Go, C#, and Visual Basic.

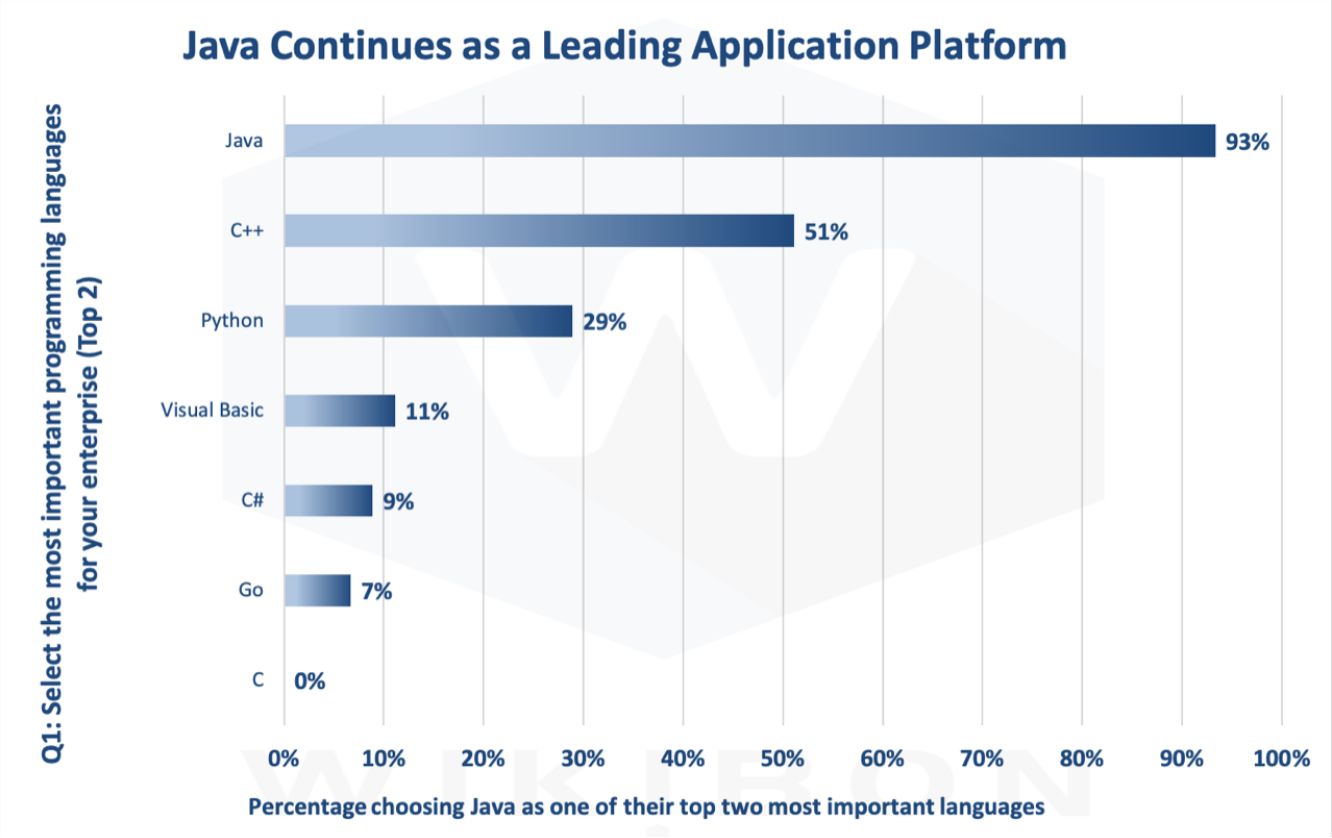

Figure 2 - Java Continues as a Leading Application Platform

Source: Wikibon © 2020. Survey of 122 enterprises with more than 1,000 employees in 2019 and 10 in-depth interviews in 2019 and 2020. Selection is Employees ≥1,500 ≤10,000, Java Applications ≥ 18, n=45.

These results illustrate that Java continues as a leading application platform. 93% of the respondents regarded Java as one of their top two programming languages, significantly ahead of C++ (51%) and Python (29%).

Mid-sized enterprises selecting Java as a top 2 programming language is higher than the Global 2000 (G2K) customers analyzed in our previous report (83%). The priority of other platforms is similar for very large and mid-sized enterprises.

We conclude that mid-sized enterprises focus resources on fewer platforms. The survey, in general, confirms that Java is essential to developers and ISVs across all industries, enterprise sizes, and government agencies (Figure 7 in the Footnotes section shows the details of that distribution).

JAVA SHORTFALLS

Figure 3 below focuses on the answers to Question 32 (Q32).

- Q32: "In the past 2 years, have you had any of the following occur with one of your Java SE-based applications? Check all that apply."
- The choices were:
  - ❍ Unauthorized intrusion into the application, Compromised customer data or IP loss, Slow application response time, Ransomware, Inconsistent application response time, and Missing transactions.
  - ❍ Respondents selected an average of 1.9 responses.

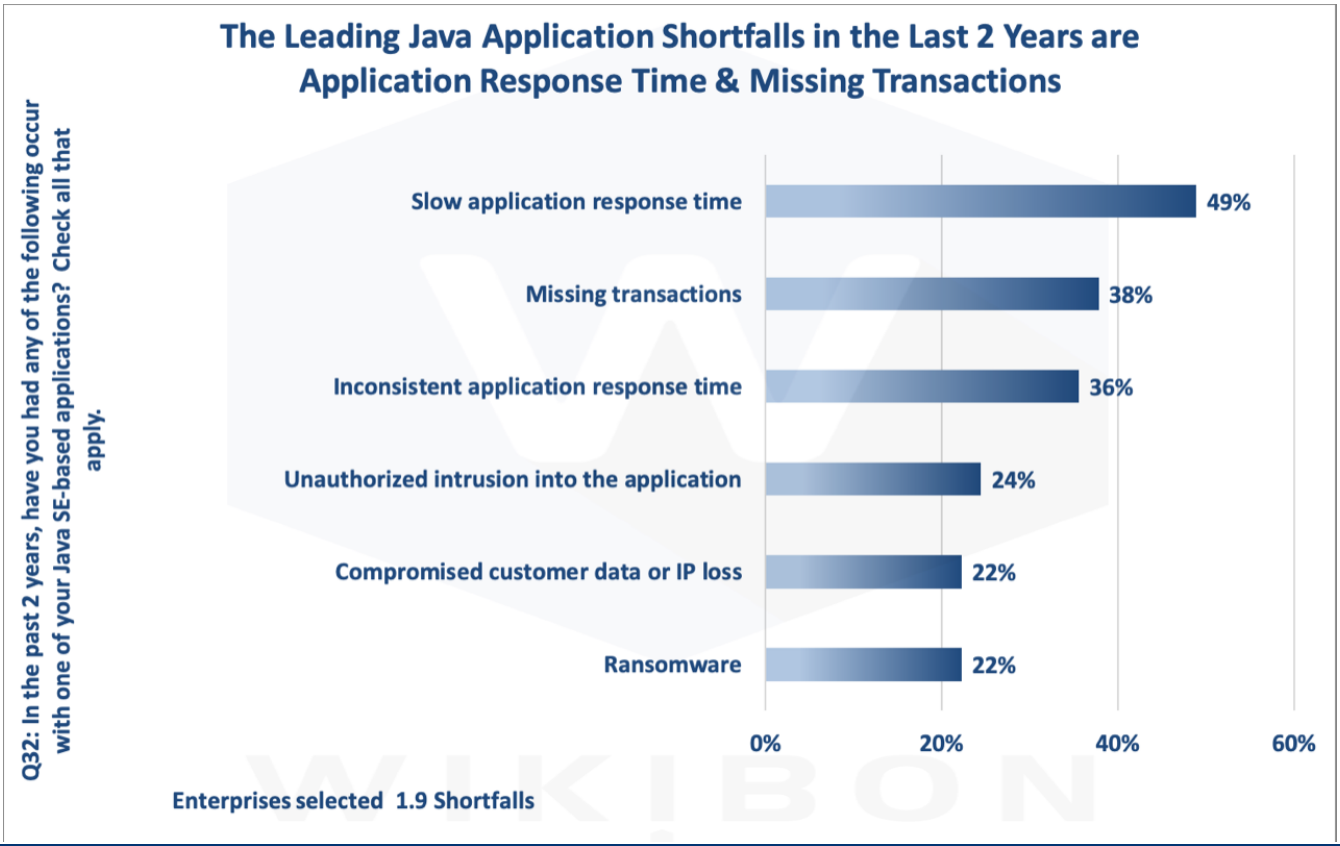

Figure 3 - Leading Java Shortfalls were Slow Application Response Time & Missing Transactions

Source: Wikibon © 2020. Survey of 122 enterprises with more than 1,000 employees in 2019 and 10 in-depth interviews in 2019 and 2020. Selection is Employees ≥1,500 ≤10,000, Java Applications ≥ 18, n=45.

Figure 3 above shows that the leading application shortfalls for Java in mid-range enterprises related primarily to application response time. The top three (3) are slow application response time, missing transactions, and inconsistent application response times.

While these mid-sized Java users had many of the same concerns as Global 2000 Java users, they were less concerned about ransomware than their larger peers. Wikibon expects this will change significantly over the next few years.

DRIVERS FOR JAVA UPGRADES

Figure 4 below focuses on the answers to Question 11 (Q11).

- Q11: "What are the most important reasons for upgrading your Java SE–based desktop or server applications with new versions of Java SE? Select up to 3."
- The choices were:
  - Better application performance, We want to stay current with latest features, More reliable performance, Better application stability, Important customers demand it, Compliance and regulatory requirements.
  - Respondents selected an average of 2.8 responses.

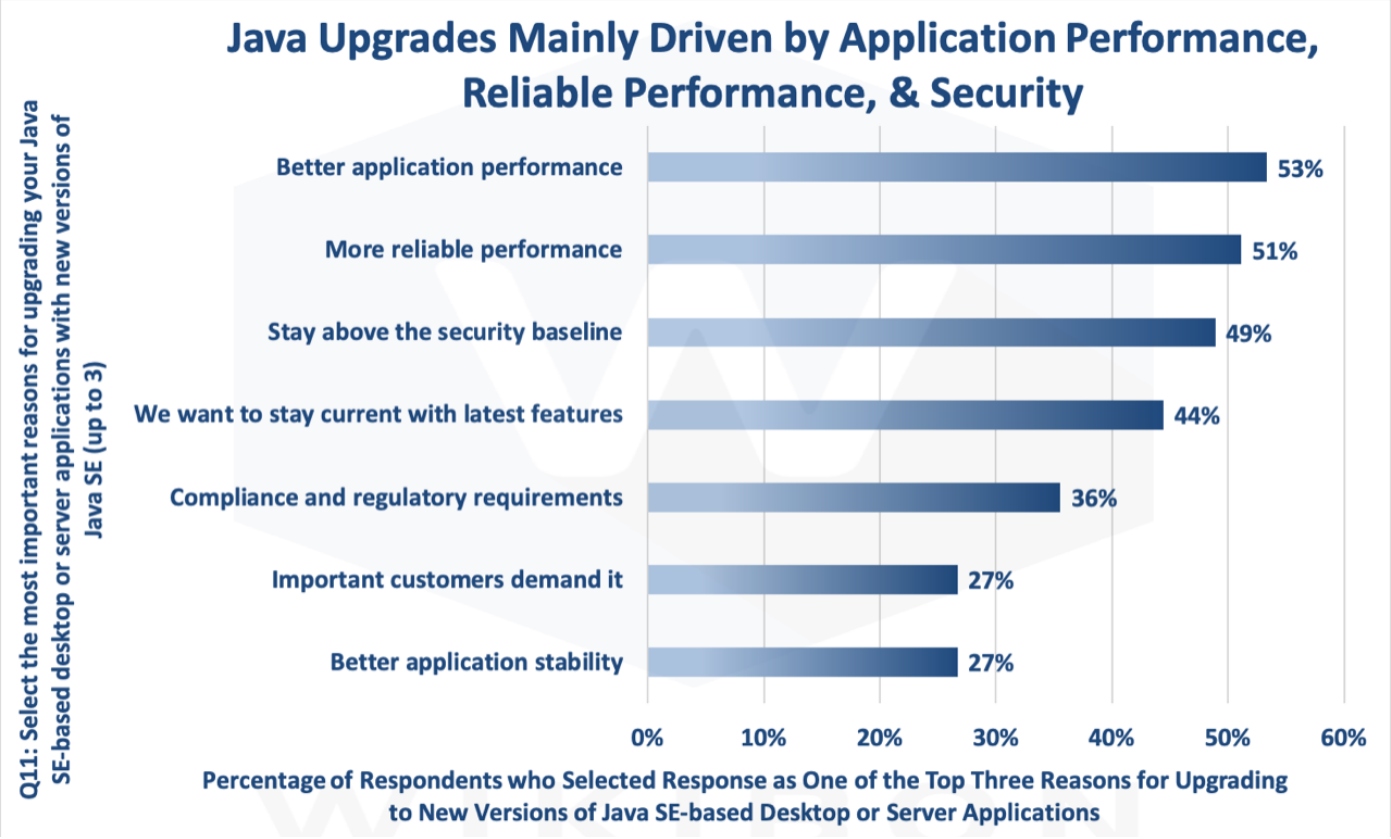

Figure 4 - Better Application Performance, More Reliable Performance, & Security are Leading Drivers for Java Upgrades.

Source: Wikibon © 2020. Survey of 122 enterprises with more than 1,000 employees in 2019 and 10 in-depth interviews in 2019 and 2020. Selection is Employees ≥1,500 ≤10,000, Java Applications ≥ 18, n=45.

Figure 4 above shows that better application performance, more reliable performance, and security were the top three drivers of Java upgrades. Wanting to stay current with the latest features was the next driver of importance. All the features had significant votes, which suggests that the focus of the Java community is in line with mid-size enterprise requirements.

Our in-depth interviews with senior Java executives in mid-range IT departments indicated that upgrades are needed more frequently for business-critical Java applications. They also indicated that support for upgrades across the Java application portfolio needed to be more flexible.

One significant difference between the mid-size and Global 2000 users is that application stability was somewhat less important for mid-sized enterprises than for G2K enterprises with more than 30,000 employees.

JAVA SUPPORT MID-SIZE ENTERPRISE STRATEGIES

In addressing their strategies for upgrades and support, the survey data shows that mid-size enterprises use multiple approaches and support options for different parts of their Java portfolio. 69% of mid-sized enterprises have a Java SE Subscription or other support from Oracle. Although the open-source software is free, the support of open-source software has never been free. Enterprises have always paid for third party support of open-source software. Red Hat for Linux is an obvious example of this practice.

The same is true for Java, with support services being offered by Bellsoft, Oracle, Red Hat, and other vendors. Some vendors provide support via their other products, such as Red Hat, with RHEL licensing. As a result of our in-depth interviews with mid-size enterprises and the survey data, Wikibon strongly recommends taking a Java portfolio view for support. We suggest that mid-size enterprises should consider their entire Java application portfolio and the different requirements of each application when making decisions about support.

Some applications require mission-critical support services, where the latest release of Java could be significant, and external support services need to be available at short notice. At the other extreme, many simple Java read-only applications may run infrequently and not need attention except when they fail.

We conclude that a workable support strategy needs a long-term model, which allows for a flexible support profile for the different applications in a broad portfolio of Java application systems. This allows each application type in the portfolio to have different requirements and timescales for upgrades, updates, and patches.

JAVA UPGRADE STRATEGIES FOR MID-SIZE ENTERPRISES

Figure 5 below focuses on the answers to Question 33 (Q33).

- Q33: "Which statement best describes your attitude about your most important Java-based applications?"
- The choices were:

❍ I prefer staying with a stable version of JDK for my most important Java SE-based applications and will tend to upgrade to new versions of Java only infrequently,

❍ I prefer staying with a stable version of Java SE for my most important Java SE-based applications, but will upgrade to new versions of Java SE on a regular cadence to take advantage of new features and functions,

❍ I try to stay as up to date as I can with Java SE versions and updates so I can take advantage of new features and functions.

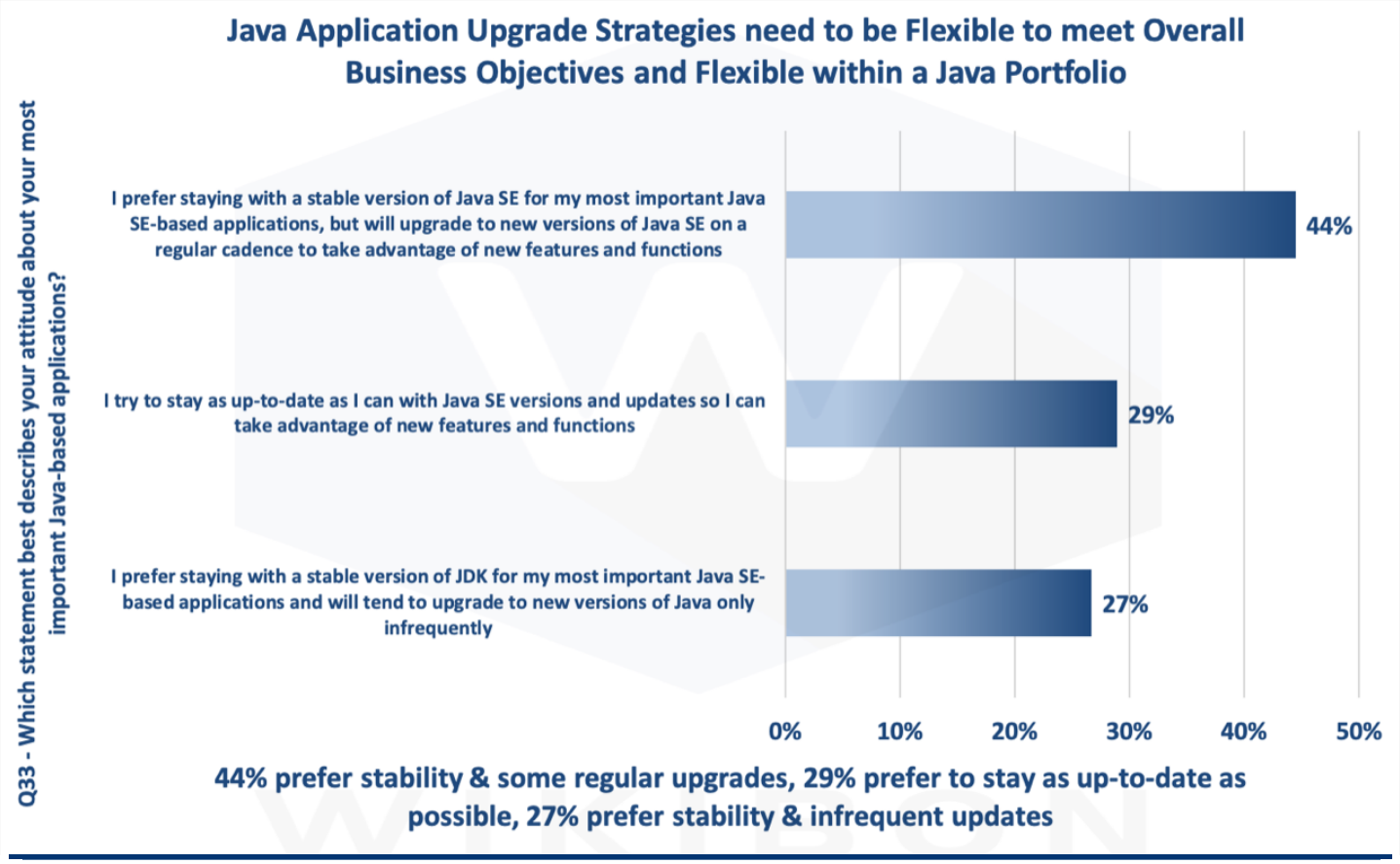

Figure 2 - Java Application Portfolio Upgrade Strategies

Source: Wikibon © 2020. Survey of 122 enterprises with more than 1,000 employees in 2019 and 10 in-depth interviews in 2019 and 2020. Selection is Employees ≥1,500 ≤10,000, Java Applications ≥ 18, n=45.

In Figure 5, respondents characterized their overall Java application portfolio upgrade strategy for key applications they employ to meet their overall business objectives. The responses indicate a diversity of strategies among mid-size Java users. Still, the most frequent approach is a preference for Java application stability, with an understanding that they will upgrade to new versions on a regular cadence to take advantage of new features and functions. This suggests 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.

One interesting difference between the upgrade strategies of G2K and mid-sized enterprises is that the latter is more aggressive in adopting the “up to date Java SE versions and updates to take advantage of new features and functions” strategy (29%) compared with only 12% of very large Java enterprises.

ORACLE JAVA SE SUBSCRIPTION FOR MID-SIZED ENTERPRISES

NEW SUPPORT OFFERINGS

The recent change to a six-month Java release cycle and the "End of Public Updates" (EoPU) for Java 8 have changed the support requirements for users, resulting in new offerings from Java support vendors. Oracle has retired its legacy and higher-cost perpetual licenses and separate support service. In its place, they have 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.

Oracle has the advantage of being the primary steward of Java and is the largest contributor to the platform. Wikibon observes that other vendors have their versions of a Java SE subscription support available 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 significant Java portfolio. The cost of this offering is predictable and mid-sized enterprises are likely to find overall savings.

Whenever there is any change in product or support offerings, every enterprise 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 business model, their skill levels, and other factors.

However, for most enterprises, Wikibon believes there is usually a 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 in more detail.

FINANCIAL ANALYSIS OF JAVA SUPPORT STRATEGY OPTIONS

The four-year financial analysis in Figure 6 below represents the likely outcome of an IT business case for an enterprise with revenues of $2.5B, an IT budget of $87M, and a total IT staff of 286. We estimated that about 171 of the IT staff would be supporting a Java portfolio of 272 applications. The source of all the detailed parameters in this model is the 2019 Wikibon survey of senior Java executives, as well as in-depth interviews with 10 Java business managers. Table 1 through Table 4 in the Footnotes below list the detailed data points from the survey, the assumptions, and calculations.

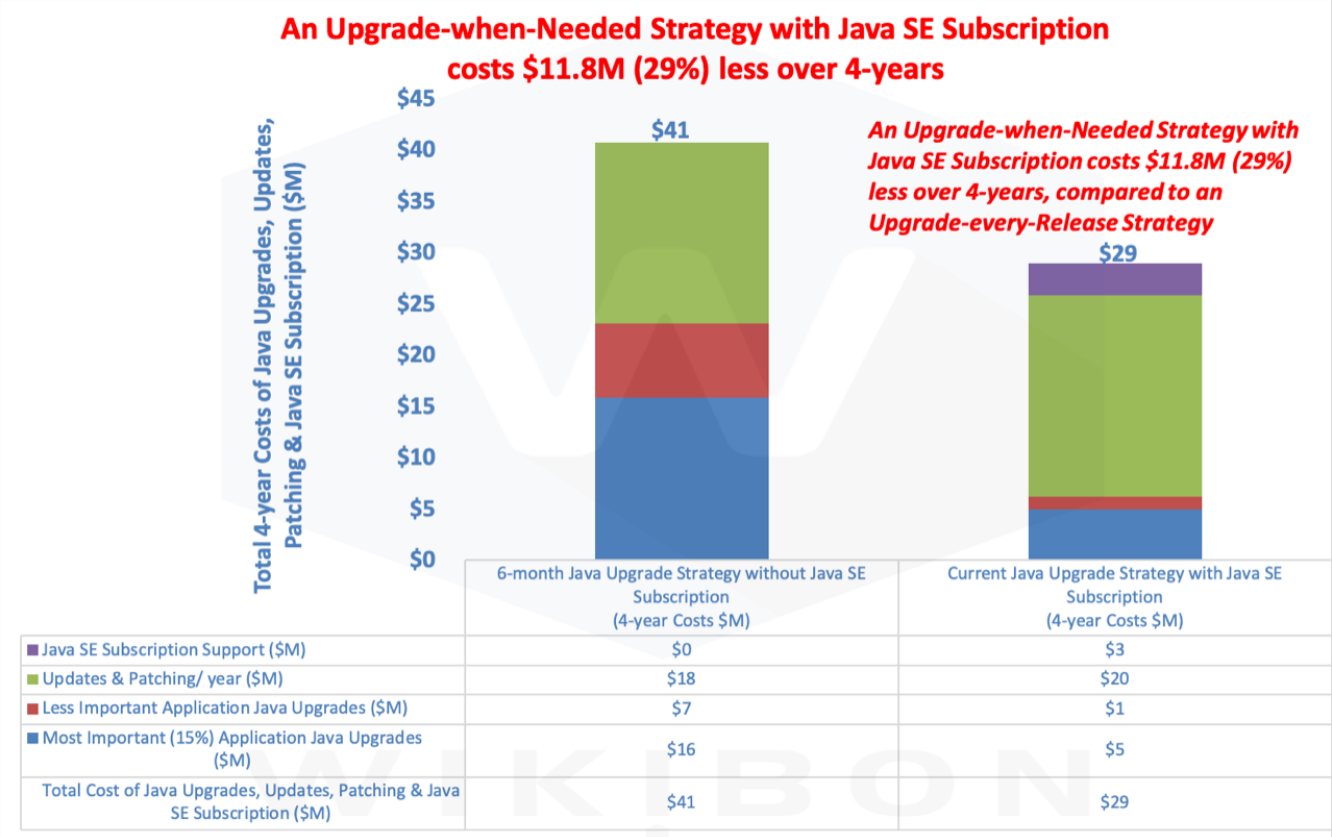

Figure 6 - Comparing "Upgrade when Needed" vs. "Upgrade every Release" (Every Six-Months) Java Strategies.

Source: Wikibon © 2020. Survey of 122 enterprises with more than 1,000 employees in 2019 and 10 in-depth interviews in 2019 and 2020. Selection is Employees ≥1,500 ≤10,000, Java Applications ≥ 18, n=45. The data comes from Table1 though Table 4 in the Footnotes below.

The data comes from Table 4 in the Footnotes below, which is derived from Table 1 through Table 3. The left-hand column in Figure 6 shows the costs of a six-month Java upgrade strategy. This is composed of Updates and Patching ($18 million), less important Java application upgrades ($7 million), and most important Java application upgrades ($16 million). 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 $41M.

The right-hand column in Figure 6 shows lower 4-year costs ($29M) for an "upgrade when needed" strategy supported by the Java SE subscription service. This is composed of Updates and Patching ($20 million), less important Java application upgrades ($1 million), and most important Java application upgrades ($5 million). 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.

In addition to this flexibility, our research shows that the Java SE Subscription is likely to be 29% less expensive than a six-month upgrade strategy because of lower labor requirements. Beyond the cost savings of $11.8 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.

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 benefits from the flexibility and speed-to-resolution offered by a Java subscription service as well.

JAVA ROADMAP

JAVA CHALLENGES

Executive management of all enterprises is focused on eliminating IT work that can be outsourced to cloud providers, either private on-premises clouds or public clouds. Cloud-native applications are built around microservices that are specific to a specific cloud.

Oracle needs to be relevant to all potential users. If the user is in a smallish company and wants to grow and go to the cloud, he will be asked the question, "Will Oracle Java be an issue?" Oracle is quietly building the Java platform as relevant to all users. Wikibon believes Oracle has a vision and the scale to get Java users to the cloud and for Java to perform more efficiently with a lighter footprint and lower cost in the cloud. Wikibon observes that Oracle needs to accelerate its initiatives in these areas, both for its own OCI and other cloud initiatives.

To be successful, Java will need to embrace a volume strategy rather than optimize revenue in the short-term. Java and Oracle will need to be relevant to ISV developers in particular, and tap into the expertise available in this community more effectively.

JDK ENHANCEMENT PROCESS

The JDK evolves through a process where JDK Enhancement Proposals (JEPs) are collected, reviewed, and tracked at OpenJDK. In this place, collaboration on an open-source implementation of Java SE primarily occurs.

The primary goal of this process is to produce a regularly-updated list of proposals to serve as the long-term Roadmap for JDK Release Projects and related efforts. The Roadmap should extend at least three years into the future to allow sufficient time for the most complex proposals to be investigated, defined, and implemented.

The last five releases and the expected September 2020 release are all exactly 6-months apart. A summary of the content in the releases is shown below.

- JDK 10
  - JDK 10 was released on March 19, 2018.
- JDF 11
  - JDK 10 was released on September 17, 2018.
- JDK 12
  - JDK 12 was released on March 19, 2019. Among others, Java 12 includes a number of new features, such as Shenandoah (A Low-Pause-Time Garbage Collector (Experimental)).
- JDK 13
  - JDK 13 was released on September 17, 2019. Java 13 includes three (3) new features and two (2) preview features (Switch Expressions and Text Blocks).
- JDK 14
  - JDK 14 was released on March 17, 2020. Java 14 includes ZGC on MacOS and Windows, and a large number of other features
- JDK 15
  - JDK 15 is planned for release on September 15, 2020. Some of the JEPs currently targeted for JDK 15 include ZGC: A Scalable Low-Latency Garbage Collector (Production) and Shenandoah: A Low-Pause Time Garbage Collector (Production).

There are several OpenJDK projects which are specifically dedicated to improving Java performance in private and public cloud environments. These improvements are aimed at lowering the cost and improving the speed of running Java applications. The following are some of the important OpenJDK projects which are part of the 3-year roadmap.

ZGC OPENJDK PROJECT

Many Java applications are designed to serve thousands or even millions of concurrent users. These applications need to allocate huge amounts of DRAM. Managing all that memory can easily impact application performance. Therefore, it is important for Java to reclaim memory dynamically and efficiently.

The objective of the ZGC project is to create a scalable, low latency garbage collector capable of handling large heaps. This is separate from the alternative experimental project Shenandoah, which is a low-pause time garbage collector. The ZGC project has been GA, and the default GC for Oracle Java releases since Java 11. Improvements

and innovations continue in subsequent releases, and performance improves towards the goal for ZGC to be a sub-millisecond maximum pause time Garbage Collector, where pause times do not increase with heap, live-set, or root-set size.

PANAMA OPENJDK PROJECT

The Panama project has the objective of providing higher performance and easier development of I/O intensive applications through the ability to more easily interface with native libraries and applications. These enhancements will allow programmers to access functionality outside of the JDK libraries such as OpenGL. This capability will also enable Java applications to interact with legacy code and systems.

VALHALLA OPENJDK PROJECT

Today's object references often point to data values spread out inefficiently throughout memory. The Project Valhalla enhancements introduce Value Types, which are designed to enable a list or array of values to be laid out linearly in a consecutive block of memory. This should lead to fewer cache misses and improved performance of machine learning and big data applications.

LOOM OPENJDK PROJECT

In order for massive scaling of applications, lightweight threads are required to make concurrency simple for programmers. Project Loom proposes user-mode threads that rely on a Java runtime implementation of continuations and schedulers instead of being implemented in the OS. This will make writing asynchronous applications, a popular programming style, much easier for developers.

AMBER OPENJDK PROJECT

The aim of the Amber project is to continuously improve developer productivity for Java developers by making code easier to write, read, and maintain. In other words, its goal is to enable cleaner code with less boilerplate.

GRAAL PROJECT

GraalVM is a low-overhead high-performance runtime written in Java. The functionality focuses on reducing startup times, improving performance, reducing CPU use, and lowering memory consumption. In particular, it allows native GraalVM images to be generated for known environments before execution. An example of the potential for performance improvement is illustrated by an Oracle benchmark showing that the startup time when running the Micronaut application on the JVM is 1202ms. With a native GraalVM image, the startup time is 18ms, a 60X improvement.

GraalVM has a cost profile similar to Java SE Subscription, and Java SE Subscription is a pre-requisite for GraalVM. GraalVM is included in the cost of Oracle Cloud Infrastructure (OCI). It is also available on-Premises as Oracle Exadata Cloud@Customer and Dedicated Region Cloud@Customer.

WIKIBON CONCLUSIONS ON JAVA DIRECTIONS

Wikibon concludes that Java is an important platform used by a high percentage of enterprises. The release data above shows that the faster Java release pace (every six months) has been well established over the last two years. As a result, the time to availability of new function has been improved. This faster time to availability has also led to further improvements being delivered earlier, as a result of Java community earlier trials and deployment.

Java is well established in the ISV community as well as enterprise development shops. However, there are simpler platforms that are gaining traction, such as Python. Wikibon believes it is important that the Java fundamentals remain relevant to the entire development market, and do not focus just on the current Java strengths in high availability and recovery.

It is also vital that Java continues down the path of becoming a fully autonomous platform that will minimize the IT effort required for patching, updates, and upgrades. It needs to develop even further and become a platform that can be used by user-department business analysts directly.

CONCLUSIONS: JAVA WILL CONTINUE AS A LEADING APPLICATION PLATFORM ON-PREMISES AND IN THE CLOUD OVER THE NEXT DECADE

The first conclusion is that Java is alive and well, and is the platform of choice for mission-critical high-value systems for mid-size enterprises. Wikibon believes that the change to a six-month release cadence is essential 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. Oracle Java has a strong development roadmap that is aimed at improving performance and manageability. The open-source nature of the project makes it natural that Java will continue to be available on multiple cloud platforms.

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 unlikely.

The financial analysis illustrated in Figure 6 shows that costs are significantly lower (29%) if an enterprise avoids upgrading every six months and deploys the Oracle Java SE Subscription model. This is a more flexible and lower cost ($12 Million less) way of balancing the upgrade, update, and patching requirements within an application portfolio.

In summary, Java has prospered well for 25 years (as of August 25, 2020). Wikibon 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, in-depth discussions, and financial analysis, Wikibon 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 parts of 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.

There are important OpenJDK projects that will enable Java applications to run more efficiently in private on-premises and public cloud environments. Wikibon recommends close monitoring of these projects to ensure they can be incorporated as necessary.

FOOTNOTES

JAVA INDUSTRY SURVEY DISTRIBUTION FOR MID-SIZED JAVA ENTERPRISES

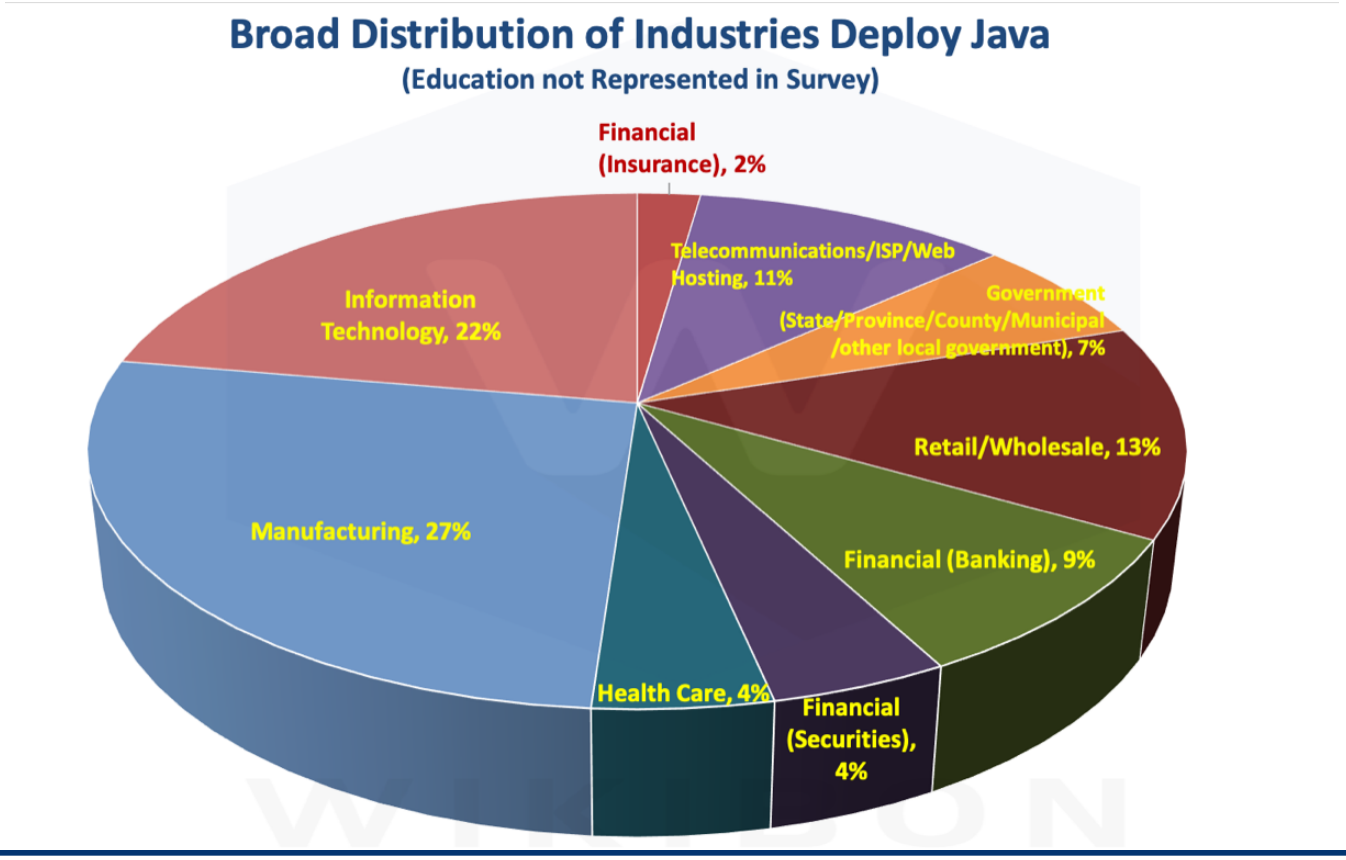

Figure 7 - Very Broad Distribution of Industries Deploy Java

Source: Wikibon © 2020. Survey of 122 enterprises with more than 1,000 employees in 2019 and 10 in-depth interviews in 2019 and 2020. Selection is Employees ≥1,500 ≤10,000, Java Applications ≥ 18, n=45.

DETAILED CALCULATIONS AND ASSUMPTIONS USED IN THE FINANCIAL MODEL

The charts and tables below are the source of Figure 1 and the Financial Analysis of the Java Support Strategy Options section and Figure 6 above.

FOOTNOTES CONTINUED ON NEXT PAGE

Sources	Current Stable Enterprise Java Environment using Java SE Support - Base Business Metrics & Java Platform Resources	
Survey QS2 Employees ≥ 1,500 & ≤ 10,000	**Number of Employees (or Equivalent)**	**8,268**
Wikibon CIO Surveys	**Revenue per Employee**	**$300,000**
Calc	**Total Revenue ($Billion)**	**$2.5**
3.5% Revenue - Average IT Environment	**IT Budget ($Million)** - Average level of IT investment at 3.5% of Revenue	**$87**
Salary $90K, 35% overhead, 40% IT Budget	**Internal IT Staff**	**286**
	Number of Java Apps in Portfolio	
Survey Q3	*Java 8 or earlier*	***83***
Survey Q4	*Java 11 (LTS release)*	***84***
Survey Q5	*Java 9, 10, 12+ (non LTS releases)*	***105***
Survey Q2 Java Apps ≥ 18	**Total Number of Java Apps**	**272**
Survey Q7&8	***Internal Staff focused on Java Platform, Patches, Updates, & Upgrades***	***96***
Survey Q7&8	***Internal Staff focused on New Java App Development & Java App Extensions***	***75***
	Total Number of Internal Java Staff	**171**
Calc	**Other Internal IT Staff**	**114**
Survey Q12&13	**Internal Resources for Java Upgrade for 15% Most Important Java Applications (Person Years)**	**15**
Wikibon Estimate 1/2 resources for 85% = 2.8x	**Internal Resources for Java Upgrade for 85% Less Important Java Applications (Person Years)**	**43**
Survey Q15&16	**External Resources for Java Upgrade for 15% Most Important (Person Years)**	**5**
Wikibon Estimate 1/2 resources for 85% = 2.8x	**External Resources for Java Upgrade for 85% Less Important Java Application (Person Years)**	**13**
Survey Q24	**Internal Resources for Updating & Patching Java Applications (Person Years/ Year)**	**38**
Survey Q27	**External Resources for Updating & Patching Java Applications (Person Years/ Year)**	**12**
Survey Q9 + Q10	**Time between Most Important Java Application Upgrades (Months)**	**21**
Survey Q9 & Q10 & Wikibon Analysis	**Time between Less Important Java Application Upgrades (Months)**	**40**
Survey Q17	**Elapsed Time to Complete Java Application Upgrade (Months)**	**4.5**

Table 1 - Wikibon Java Survey Results and Calculations for Mid-sized Enterprises

Source: Wikibon © 2020. Survey of 122 enterprises with more than 1,000 employees in 2019 and 10 in-depth interviews in 2019 and 2020. Selection is Employees ≥1,500 ≤10,000, Java Applications ≥ 18, n=45. See Table 5 for the text of survey questions reference in the source column.

Source	Current Stable Enterprise Java Environment using Java SE Support	
	Cost of Java Version Upgrade for Most Important Applications (15% of 272 Java Apps = 41)	
a	Number of Internal Person Years for Upgrading Most Important Applications	15.3
b	Cost of Upgrade Persons (including 35% overhead)	$121,500
c	Number of External Person Years for Upgrading Most Important Applications (Consultant)	1.4
d	Cost of External Specialist Consultant Upgrade Persons	$190,000
e	Number of External Person Years for Upgrading Most Important Applications (Outsourced)	3.2
f	Cost of External Outsourced Upgrade Persons	$30,375
g = (a x b) + (c x d) + (e x f)	**Total Cost of Upgrade of Most Important Applications**	**2,219,225**
h	Time between Upgrades (Months)	21
j = g x 12 / h	**Total Cost of Java Upgrades for Most Important Applications/ Year**	**$1,240,561**
	Cost of Java Version Upgrade for Less Important Applications (85% of 272 Java Apps = 231)	
k	Number of Internal Person Years for Upgrading Less Important Applications	43.4
b	Cost of Upgrade Persons (including 50% uplift for overheads)	$121,500
m	Number of External Person Years for Upgrading Less Important Applications (Consultant)	3.9
d	Cost of External Specialist Consultant Upgrade Persons	$190,000
n	Number of External Person Years for Upgrading Less Important Applications (Outsourced)	9.1
f	Cost of External Outsourced Upgrade Persons	$30,375
p = (k x b) + (m x d) + (knx f)	**Total Cost of Upgrade of Less Important Applications**	**$1,015,567**
q	Time between Upgrades (Months)	40
r = p x 12 / q	**Total Cost of Java Upgrades for Less Important Applications/ Year**	**$304,670**
s	Number of Internal Person Years for Updates & Patching/ Year	38
b	Cost of Internal Staff for Updates & Patching / Year	$121,500
t	Number of External Person Years for Updates & Patching/ Year	12
f	Cost of Outsourced External Staff for Updates & Patching / Year	$30,375
u = (s x b) + (t + f)	**Total Cost Java Updates & Patching/ year (External & Internal)**	**$4,914,602**
= a + k + s	**Number of Internal Persons Years for Upgrades, Updates, & Patching/ Year**	**59**
= c + e + m + n + t	**Number of External Person Years for Updates & Patching/ Year**	**29**
= j + r + u	**Total Cost of Maintaining Java Performance, Functionality, Security, & Compliance/ Year**	**$6,459,833**
= (a + k + s)/ Survey Q8	**Percentage of Internal Java Staff Dedicated to Java Upgrades, Updates & Patching**	**34%**
= (j + r + u)/IT Budget]	**Percentage of Total IT Budget**	**7.4%**

Table 2 - Wikibon Java Survey Results and Calculations for Mid-sized Enterprises

Source: Wikibon © 2020. Survey of 122 enterprises with more than 1,000 employees in 2019 and 10 in-depth interviews in 2019 and 2020. Selection is Employees ≥1,500 ≤10,000, Java Applications ≥ 18, n=45. See Table 5 for the text of survey questions reference in the source column.

Source	Impact on Enterprise Java Environment Implementing an Upgrade Every Six Months Strategy	
	Cost of Java Version Upgrade for Most Important Applications (15% of 272 Java Apps = 41)	
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	10.2
b	Cost of Upgrade Persons (including 35% overhead)	$121,500
= c / v	Number of External Person Years for Upgrading Most Important Applications (Consultant) / Reduction Factor	0.9
d	Cost of External Specialist Consultant Upgrade Persons	$190,000
= e / v	Number of External Person Years for Upgrading Most Important Applications (Outsourced)	2.1
f	Cost of External Outsourced Upgrade Persons	$30,375
as above	**Total Cost of Upgrade of Most Important Applications**	**$1,479,483**
as above	**Time between Upgrades (Months), determined by when support ends**	**4**
as above	**Total Cost of Java Upgrades for Most Important Applications/ Year**	**$3,955,054**
	Cost of Java Version Upgrade for Less Important Applications (85% of 272 Java Apps = 231)	
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	28.9
b	Cost of Upgrade Persons (including 50% uplift for overheads)	$121,500
= m / v	Number of External Person Years for Upgrading Less Important Applications (Consultant) /Reduction Factor	2.6
d	Cost of External Specialist Consultant Upgrade Persons	$190,000
= n / v	Number of External Person Years for Upgrading Less Important Applications (Outsourced) / Reduction Factor	6.1
f	Cost of External Outsourced Upgrade Persons	$30,375
as above	**Total Cost of Upgrade of Less Important Applications**	**$677,045**
as above	**Time between Upgrades (Months), determined by when support ends**	**4**
as above	**Total Cost of Java Upgrades for Less Important Applications/ Year**	**$1,809,921**
w	Saving in Updates & Patching after new updates	10%
= s x (1 - w)	Number of Internal Person Years for Updates & Patching/ Year	34
b	Cost of Internal Staff for Updates & Patching / Year	$121,500
= t x (1 - w)	Number of External Person Years for Updates & Patching/ Year	11
f	Cost of Outsourced External Staff for Updates & Patching / Year	$30,375
as above	**Total Cost Java Updates & Patching/ year (External & Internal)**	**$4,423,142**
as above	**Number of Internal Persons Years for Upgrades, Updates, & Patching/ Year**	**138**
as above	**Number of External Person Years for Updates & Patching/ Year**	**23**
as above	**Total Cost of Maintaining Java Performance, Functionality, Security, & Compliance/ Year**	**$10,188,118**
As above	**Percentage of Internal Java Staff Dedicated to Java Upgrades, Updates & Patching**	**81%**
As above	**Percentage of Total IT Budget**	**11.7%**

Table 3 - Calculations of Line items in Mid-size Enterprise Java Environment using an Upgrade Every Six-months Strategy.

Source: Wikibon © 2020. Survey of 122 enterprises with more than 1,000 employees in 2019 and 10 in-depth interviews in 2019 and 2020. Selection is Employees ≥1,500 ≤10,000, Java Applications ≥ 18, n=45.

FOOTNOTES CONTINUED ON NEXT PAGE

Comparative Four-year Cost of Java Upgrades, Updates & Patching	
Current Java Upgrade Strategy with Java SE Subscription	**4-year Costs**
Cost of Most Important Application Java Updates	$4,962,242
Cost of Less Important Application Java Updates	$1,218,680
Cost Java Updates & Patching/ year (External & Internal)	$19,658,410
Cost of Oracle Java SE Subscription Support (or Equivalent Enterprise Agreement)	$3,077,000
Total Cost of Java Upgrades, Updates, & Patching with Current Upgrade Strategy	**$28,916,332**
6-month Java Upgrade Strategy without Java SE Subscription	**4-year Costs**
Cost of Most Important Application Java Updates	$15,820,217
Cost of Less Important Application Java Updates	$7,239,686
Total Cost Java Updates & Patching/ year (External & Internal)	$17,692,569
Cost of Oracle Java SE Subscription Support (or Equivalent Enterprise Agreement)	$0
Total Cost of Java Upgrades, Updates, & Patching with 6-month Upgrade Strategy	**$40,752,472**
Total 4-year Savings from Use of Java SE Subscription Support	**$11,836,140**

Table 4 - Summary of 4-year IT Budget Savings of using Java SE Subscription vs. Implementing a 6-month Upgrade Strategy for Mid-size Enterprises.

Source: Wikibon © 2020. Survey of 122 enterprises with more than 1,000 employees in 2019 and 10 in-depth interviews in 2019 and 2020. Selection is Employees ≥1,500 ≤10,000, Java Applications ≥ 18, n=45. See Table 5 for the text of survey questions reference in the source column.

Wikibon Survey Questions Referenced in Table 1	
QS2	How many total employees work at your enterprise?
Q2	How many Java-based applications are running in your enterprise or division?
Q3	Most companies run a combination of Java SE versions. Characterize the amount of usage of each version in your enterprise or division.
Q4	Most companies run a combination of Java distributions. Characterize the amount of usage of each distribution in your enterprise or division.
Q5	What is the latest Java version that you are evaluating/considering now (even if you have not deployed it yet)?
Q7	How many Java developers does your enterprise have?
Q8	Roughly, what proportion of your Java SE developers are · IT-focused Java specialists who are primarily responsible for monitoring and evaluating Java upgrades and releases? _____% · Tactical and/or strategic business application programmers who are primarily responsible for writing and maintaining Java SE applications? _____%
Q9	When did you last upgrade your Java SE version or install a new release?
Q10	Going forward, when do you expect you will next upgrade your most important Java-based applications with new versions of Java SE?
Q12	How many of your internal Java programmers (including all Java specialists, developers, and application staff) are typically involved in an upgrade (i.e. Java 8 to Java 11) of an important application?
Q13	How many days each do your Java developers spend supporting your typical upgrade?
Q15	How many external, 3rd party Java programmers are typically in involved in an upgrade?
Q16	How many days each do your 3rd party Java programmers spend supporting your typical upgrade?
Q17	How many months does a Java SE application upgrade typically take? (i.e., Java application is upgraded to execute on a new version/release of Java, for instance from Java 8 to Java 11)
Q24	How many of your internal Java SE programmers (including all Java specialists, developers, and application staff) are typically involved in updating and patching your Java SE-based applications over the course of a year?
Q27	How many external, 3rd party Java programmers are typically in involved in a typical update or patch?

Table 5 - Wikibon Java Survey Questions Referenced in the Source Column of Table 1 above.

Source: © Wikibon, 2019. Wikibon Java Survey Questionnaire.

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.