IDC PERSPECTIVE

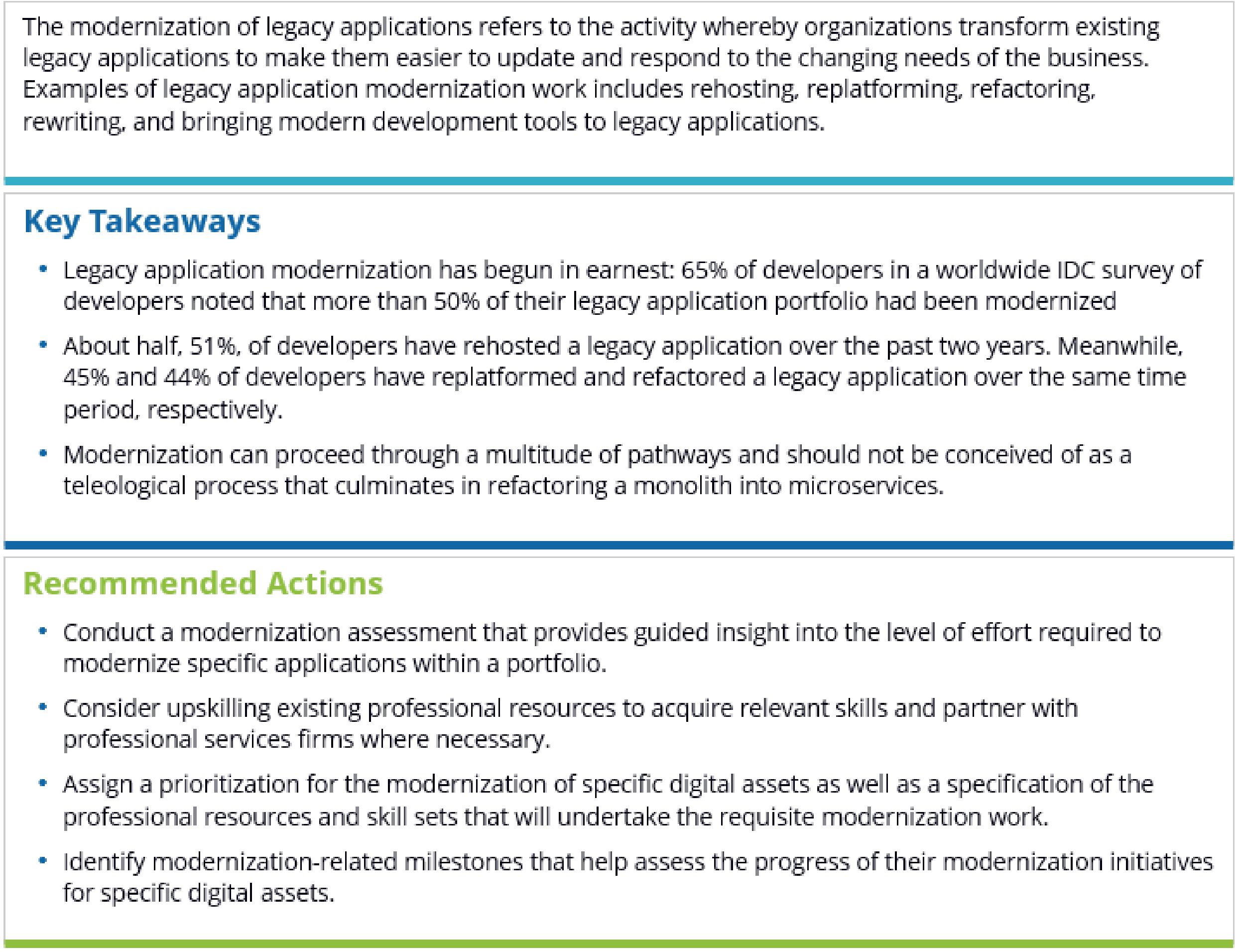
Legacy Application Modernization: Sorting Through the Options from an Application Development Perspective

Arnal Dayaratna

EXECUTIVE SNAPSHOT

FIGURE 1

Executive Snapshot: Understanding Legacy Application Modernization



The modernization of legacy applications refers to the activity whereby organizations transform existing legacy applications to make them easier to update and respond to the changing needs of the business. Examples of legacy application modernization work includes rehosting, replatforming, refactoring, rewriting, and bringing modern development tools to legacy applications.

Key Takeaways

- Legacy application modernization has begun in earnest: 65% of developers in a worldwide IDC survey of developers noted that more than 50% of their legacy application portfolio had been modernized
- About half, 51%, of developers have rehosted a legacy application over the past two years. Meanwhile, 45% and 44% of developers have replatformed and refactored a legacy application over the same time period, respectively.
- Modernization can proceed through a multitude of pathways and should not be conceived of as a teleological process that culminates in refactoring a monolith into microservices.

Recommended Actions

- Conduct a modernization assessment that provides guided insight into the level of effort required to modernize specific applications within a portfolio.
- Consider upskilling existing professional resources to acquire relevant skills and partner with professional services firms where necessary.
- Assign a prioritization for the modernization of specific digital assets as well as a specification of the professional resources and skill sets that will undertake the requisite modernization work.
- Identify modernization-related milestones that help assess the progress of their modernization initiatives for specific digital assets.

Source: IDC, 2020

SITUATION OVERVIEW

From an application development perspective, digital transformation is composed of two fundamental modalities in the form of the development of net-new applications and the modernization of legacy applications. The development of net-new applications addresses the need for the development of new applications that solve pressing business problems. Meanwhile, the modernization of legacy applications refers to the activity whereby organizations transform existing legacy applications to make them easier to update and respond to the changing needs of the business. In addition, legacy application modernization is intended to ensure the continued affordability of maintaining and managing such applications. This document examines key considerations related to the modernization of legacy applications by quantifying the progress of legacy application modernization in the application development landscape. The document also provides technology buyers with guidance about strategic planning related to legacy application modernization and best practices regarding its execution.

Definitions

A legacy application is a software application that is considered obsolete or based on outdated software or infrastructure. Legacy applications are typically critical to day-to-day operations, even though they are often not compatible with modern operating systems, hardware, and programming languages. For example, a legacy application written for Windows 7 may not be compatible with Windows 10 and subsequently requires modernization to support Windows 10 devices. Another important dimension of a legacy application is the lack of availability of talent to support its continued maintenance and management. COBOL applications written for mainframes in the financial services industry, for example, may suffer from a shortage of skilled COBOL developers even though the applications themselves are functional, are reliable, and exemplary of high performance.

What are the methods and modalities of modernizing legacy applications? And what are their relative benefits and trade-offs? There are five prominent modalities of legacy application modernization:

- **Rehosting of applications:** The lift and shift of applications from one platform to the cloud, with zero to minimal architectural changes to the application
- **Replatforming applications:** The migration of applications from an on-premises environment to a cloud, with minor changes that leverage some of the cloud platform's functionality such as autoscaling and high availability
- **Refactoring applications:** The re-architecting of applications to optimize their functionality for the cloud (One common example of refactoring involves transforming monolithic applications into microservices architectures using containers.)
- **Integration with modern development tools (IDEs, DevOps toolchains, etc.):** The use of modern development tools such as cloud-based IDEs, DevOps toolchains, APIs, microservices and containers to extend the functionality of legacy applications
- **Rewriting applications:** Rewriting and coding a legacy application from scratch

COVID-19 and Legacy Application Modernization

COVID-19's prioritization of rapid application development as a means of creating digital solutions that solve pressing problems has accelerated digitization because of the urgency of implementing digital solutions such as work-from-home infrastructures, augmented ecommerce capabilities, and contactless delivery. On one hand, the celerity of these digitization initiatives has enabled enterprises to rapidly pivot business operations to accommodate changes in the business, political, and regulatory landscape that have arisen as a result of COVID-19. On the other hand, the speed with which these solutions have been developed has often led to a corresponding failure to architect and develop solutions in ways that can continue to be used in a scalable and efficacious way for years to come, subsequent to the pandemic. Put bluntly, enterprises are creating technical debt and are subsequently amplifying the need to modernize applications as a result of their rapid development of digital solutions for COVID-19. This means that needs to modernize legacy applications will achieve increased urgency subsequent to the cessation of the pandemic because of the need to modernize applications that were rapidly created to solve pandemic-related business problems.

Legacy Application Modernization

According to *PaaSView and the Developer 2020: Key Worldwide Trends in Contemporary Application Development* (IDC #US46651020, July 2020), a recent worldwide developer survey of 2,500 developers, more than half of the application estate of enterprises are made up of legacy applications. Moreover, roughly 65% of respondents noted that more than half of their legacy application portfolio had been modernized (see in Figure 2).

Taken together, these findings reveal that more than 26% — half of 52% from the column chart on the left — have been modernized in some way (see Figure 2). While it might seem remarkable that more than half of an enterprise's portfolio of legacy applications and greater than one-fourth of its overall application estate have been subject to modernization initiatives, a deeper examination of related data reveals that the actual state of modernization initiatives worldwide is more complex. For example, data regarding the usage of modernization techniques reveals usage of discrete modernization methodologies as shown in Figure 3.

Figure 3 shows that rehosting, replatforming, refactoring, rewriting, and bringing modern development tools to the enterprise are all used to a significant degree. Even though rehosting is the most widely used modernization technique, and refactoring — because of its complexity — is the least commonly used method, all of these techniques have been used by more than 40% of respondents. This finding indicates that enterprises are leveraging a multitude of modernization methodologies in parallel, and that only a small fraction of the 26% of enterprise applications that have been modernized, have experienced refactoring to a significant degree. Meanwhile, IDC survey data shows that of the applications that have been refactored, only a subset of applications have had a significant percentage of their code refactored as shown in Figure 4.

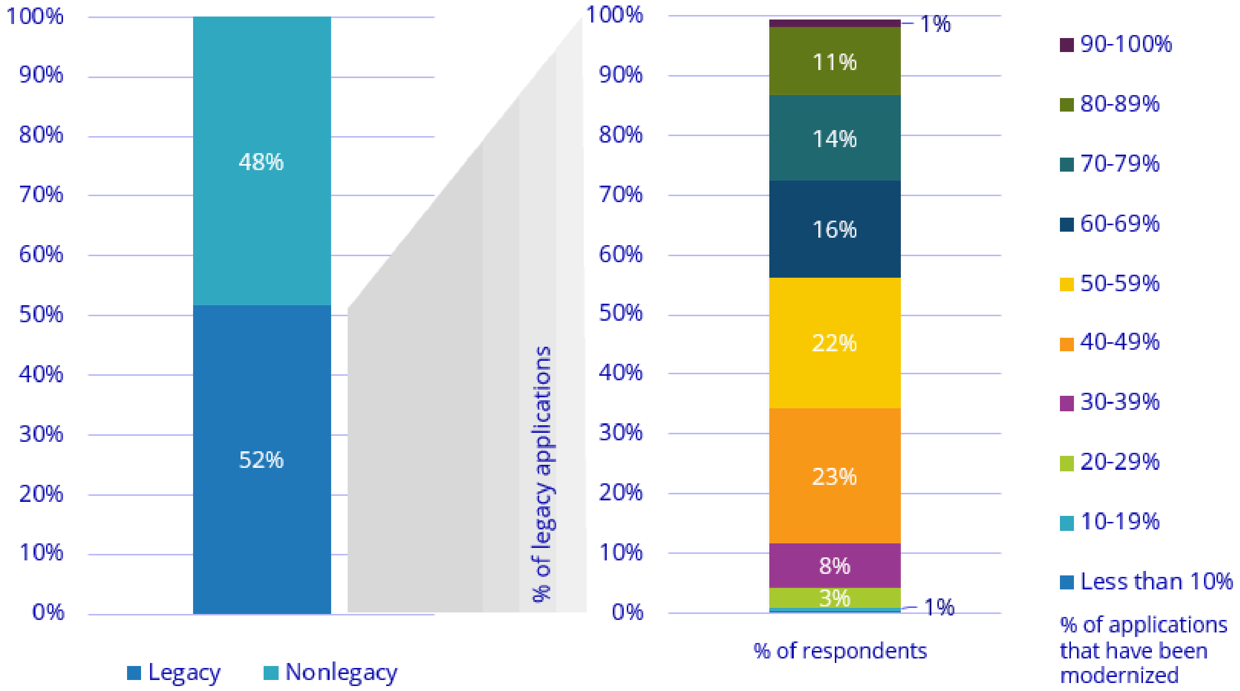
Figure 4 shows that only 1% of respondents has refactored 90–100% of an application's code and that a total of 6% respondents have refactored more than 80% of their code. As such, the chart illustrates that modernization initiatives are proceeding incrementally and that opportunities to continue refactoring application code from a monolithic application remain vast. While application modernization has begun in earnest, enterprises have significant opportunities to increase the share of legacy applications that has been modernized, the percentage of applications that has been refactored, and the percentage of application code that has experienced refactoring.

FIGURE 2

Legacy Applications That Have Been Modernized

Q. What percentage of applications in your organization are legacy versus nonlegacy?

Q. Now, what proportion of your legacy applications has been modernized?



n = 1,873

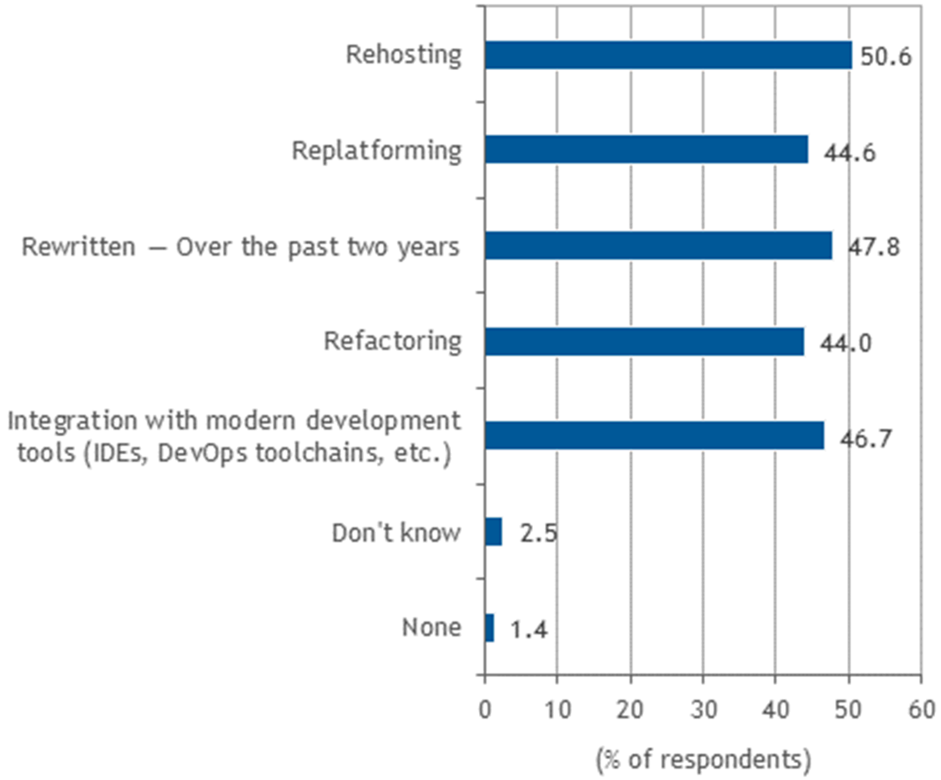
Base = all respondents

Source: IDC's *PaaSView and the Developer,* 2020

FIGURE 3

Developer Usage of Modernization Techniques

Q. Check which of the following modernization techniques you have used with respect to legacy apps over the past two years? Next two years?



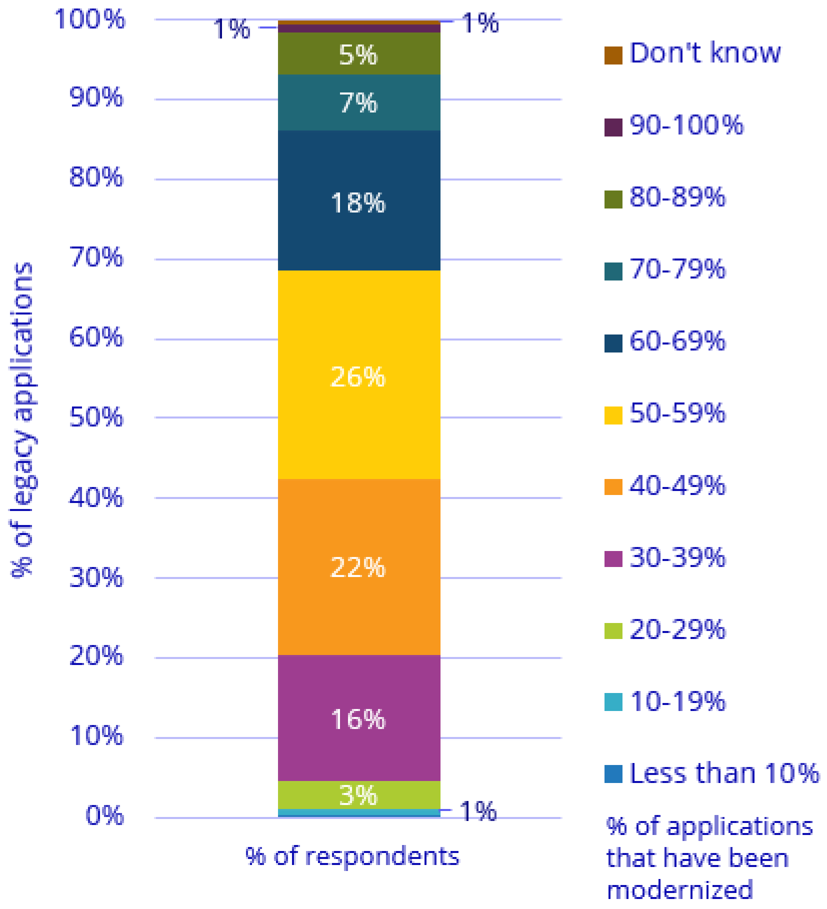
n = 1,932

Source: IDC's *PaaSView and the Developer,* 2020

FIGURE 4

Application Code That Is Refactored as Part of a Modernization Initiative

Q. *In the case of legacy applications that have been refactored, on average, what proportion of application code has been refactored into a microservices architecture?*



n = 811

Base = respondents who indicated legacy applications have been refactored

Source: IDC's *PaaSView and the Developer*, 2020

Comparing Different Modernization Modalities

Benefits and disadvantages abound for all of the modernization modalities enumeration in the Definitions section. Table 1 elaborates on the merits and demerits of discrete modernization methodologies.

TABLE 1

Comparison Between Modernization Pathways

Modernization Pathway	Benefits	Disadvantages
Rehosting	▪ This involves speed because there are minimal changes made to application architecture and applications can be rapidly migrated to the cloud. ▪ This involves reduction in on-premises expenses.	▪ Applications are not optimized for the cloud and are not able to fully benefit from their deployment to a cloud infrastructure. ▪ Applications may not be able to optimally consume cloud products and services.
Replatforming	▪ Applications can more directly benefit from cloud-specific functionality such as autoscaling and cloud-based products and services.	▪ Care must be exercised to ensure that the replatforming work does not lead to unexpected further modifications that disrupt the application as a result.
Refactoring	▪ Refactoring can transform the architecture of an application to render it easier to update for the long term. ▪ Refactoring has the potential to increase developer velocity by allowing distributed developer teams to work in parallel on application development.	▪ Refactoring is complex. There are few structured guides or templates that provide developers with insight into how to refactor applications. ▪ Refactoring can be costly and requires lengthy development cycles.
Rewriting	▪ Rewriting allows developers to bring a plethora of new features and functionality to an outdated application. ▪ Rewriting can render applications both more extensible and portable across different infrastructures.	▪ Rewriting is time consuming and costly and can also be complex. ▪ Rewriting carries it with the risk that some of the performance benefits of an existing application may be inadvertently compromised by a newer architecture.
Integration with modern development tools (IDEs, DevOps toolchains, etc.)	▪ This involves the ability to bring modern development functionality to legacy applications with minimal disruption.	▪ Because the foundational application remains untouched, there are limitations to the amount of new functionality that can be brought to a legacy application.

Source: IDC, 2020

Two concepts are worth considering when appraising the respective advantages and disadvantages of the modernization techniques enumerated previously – namely, complexity and value. Thinking about a continuum encompassed by rehosting, replatforming, refactoring, rewriting, and bringing modern development tools to legacy applications, the degree of complexity of a legacy application modernization initiative progressively decreases as we navigate "rehosting" to "bringing development tools to legacy applications." In other words, rehosting is generally considered simpler than replatforming, which is correspondingly simpler than refactoring.

With respect to the generation of value from these modernization pathways, however, there is no corresponding trend spanning these five modalities: bringing modern development tools to legacy applications, for example, can be generative of as much value, if not more, than refactoring a legacy application. The reason for this is that refactoring is often the wrong modernization approach because it can be disruptive to an application and therefore out of sync with an enterprise's modernization objectives. Even the lift and shift of a legacy application can be as generative of value as a replatforming initiative in the event that the application is already optimized for the cloud. All of this is to say that, as buyers develop a strategy that balances cost, complexity, value, and risk, careful consideration needs to be made regarding the most expeditious path to the realization of value for each individual modernization method.

ADVICE FOR THE TECHNOLOGY BUYER

Understand That Modernization Can Take Place Through Several Modalities

Modernization can take place through a multitude of discrete pathways and methodologies, and hence, is not reducible to refactoring an application or rewriting it from scratch. Rehosting and replatforming constitute two venerable modalities of modernization that are often used as part of a phased approach to modernization wherein applications are initially migrated to the cloud, and re-architected at a later stage. Benefits of rehosting and replatforming include enabling applications to leverage the benefits of the cloud. For example, applications that have been rehosted or replatformed benefit not only from the economics of cloud computing, the cloud's automated scalability and elasticity and self-service capabilities, but also from streamlined proximity to other cloud products and services.

That said, technology buyers should be careful not to think of modernization as a teleological process that begins with rehosting or replatforming and culminates in refactoring. Depending on the nature of the legacy application, refactoring is emphatically not the right approach because of considerations related to the complexity of the application and the team's modernization-related goals. The bottom line here is that modernization should be conceived as a multivalent process that leverages a constellation of methodologies, tools and infrastructures.

Understand That Modernization Is an Ongoing Process

As demonstrated by the findings that illustrate the only 1% of legacy applications has had 100% of its code refactored, modernization should be understood as a phased, gradual, and iterative process. Put differently, modernization does not involve some radical transubstantiation of a legacy application into a modern application, but rather features the sustained application of modernization processes and tools to legacy applications. As such, technology buyers would do well to identify modernization-related milestones that help them assess the progress of their modernization initiatives for specific digital assets. Another benefit of modernization milestones is their ability to provide guidance as to when the next phase of modernization should be initiated. IT leaders should consider automating the incremental execution of modernization strategies to portfolios of applications. Such automation might be as simple as integrating project management tools with existing modernization initiatives and subsequently automating the delivery of notifications and messages regarding the need to progress to the next phase of a modernization initiative.

Recognize That Modernization Is Not Necessarily About Decomposing a Monolithic Application into Microservices

While modernization is often thought of in the context of refactoring or replatforming, the activity of bringing modern development tools to a legacy application can accomplish many of the same objectives traditionally associated with modernization. For example, the use of cloud-based IDEs and APIs has the potential to extend the functionality of legacy applications by enabling the modernization of application front ends, for example, or otherwise facilitating the development of net-new functionality that complements the original application.

For example, Zowe, the open source framework for connecting applications to mainframe-based data and applications, enables the modernization of mainframe-based z/OS applications. Zowe's democratization of mainframe-based development, its improvement of developer productivity, and catalyzation of innovation enable developers to bring modern application development tools and methodologies to the enterprise. Developers can leverage Zowe to access developer tools that empower them to modernize the front end of mainframe-based applications or use APIs to enhance the functionality of existing mainframe-based products and services. In addition, developers can use Zowe to bring CI/CD tooling and increased automation to the application delivery pipeline, more generally.

Similarly, Progress OpenEdge 12, an application development platform that specializes in the development of transactional business applications, empowers developers to modernize its legacy applications by means of its support for cloud-based IDEs, compatibility with DevOps toolchains, and support for microservices, containers, and cloud-based deployments. Developers can use OpenEdge's compatibility with VS Code and Eclipse Che to create rich UX front-end applications for OpenEdge applications, or otherwise to automate the release of applications updates using CI/CD tooling.

Develop a Strategic Road Map for Modernization

Technology buyers would do well to create a strategic vision for modernization within their organization that specifies which digital assets need to be modernized, and correspondingly, what modernization method is most appropriate for the asset in question. This strategic vision needs to assign a prioritization for the modernization of specific digital assets as well as a specification of the professional resources and skill sets that will undertake the requisite modernization work. In cases where the skill sets for modernizing legacy applications are in short supply, technology buyers should consider upskilling existing professional resources to acquire relevant skills and partner with professional services firms where necessary. Many professional services firms use an operating model that involves transferring skill sets and knowledge to employees of their client organizations and, thereby, serves as auxiliary training resources for the purpose of upskilling existing developers.

Conduct a Modernization Assessment

IDC recommends that technology buyers perform a systematic assessment of a portfolio of legacy applications that examines attributes such as the following:

- Application complexity
- Application architecture
- Security vulnerabilities
- Business benefit of modernization
- Technical complexity of specific modernization pathways
- Modernization-related risks

The goal of the assessment should be to provide guided insight into the level of effort required to modernize specific applications within a portfolio. An application assessment typically leverages static code analysis tools that appraise the architecture, complexity, and security vulnerabilities within an application. In conjunction with a static code analysis, buyers would do well to assess the relevance, cost and complexity of discrete modernization methodologies as a means of making informed and strategic decisions about the best path forward for their legacy application portfolio.

LEARN MORE

Related Research

- *Broadcom's CA Brightside Brings Enterprise-Grade Support for Developer Tools That Bring Modern Application Development to z/OS* (IDC #US46765520, September 2020)
- *PaaSView and the Developer 2020: Key Worldwide Trends in Contemporary Application Development* (IDC #US46651020, July 2020)
- *The Path Toward a Cloud-Native Enterprise: PaaS, Cloud-Native Technologies, DevOps, and Developer Centricity* (IDC #US46538720, June 2020)
- *Worldwide Development Languages, Environments, and Tools Market Shares, 2019: COVID-19 Shifts Focus to Developer Agility and Rapid Application Development* (IDC #US46529620, June 2020)
- *Worldwide Development Languages, Environments, and Tools Forecast, 2020-2024* (IDC #US46529720, June 2020)
- *Red Hat OpenShift 4.2 Delivers Bevy of Developer Tools That Enhance Cloud-Native Development* (IDC #US45668219, December 2019)
- *Key Considerations for Application Transformation and Modernization Using Microservices* (IDC #US45714619, December 2019)
- *SpringOne Platform 2019: Pivotal Strengthens Its Cloud-Native Application Development Stack with Automation Atop Kubernetes and Partnership with Microsoft on Azure Spring Cloud* (IDC #US45649919, November 2019)
- *Pivot Table: Worldwide Developer Forecast, 2019-2024* (IDC #US45579719, October 2019)
- *Cultivating Mindshare Among Cloud-Native Developers* (IDC #US44493619, September 2019)
- *VMware Modernizes Its Approach to Kubernetes, Increasing Its Appeal to Developers and DevOps* (IDC #lcUS45490519, September 2019)

- *Cloud-Native Development and Applications: Notes Toward a Definition* (IDC #US45001119, April 2019)
- *Modern Development: New Developer Personas and the Changing Face of UX* (IDC #DR2019_T4_AD, March 2019)
- *The Cloud Foundry Foundation Announces Deeper Kubernetes Integration* (IDC #lcUS43408218, October 2018)
- *Pivotal Container Service Delivers Cloud-Native Solutions to the Enterprise Enhanced by VMware* (IDC #lcUS43589018, February 2018)

Synopsis

This IDC Perspective examines key considerations related to the modernization of legacy applications by quantifying the progress of legacy application modernization in the application development landscape. The document also provides technology buyers with guidance about strategic planning related to legacy application modernization and best practices regarding its execution.

"Legacy application modernization requires a multivalent and highly strategic approach to ensure that enterprises achieve their modernization goals," said Arnal Dayaratna, research director, Software Development, IDC. "Buyers should take the time to carefully conduct modernization assessments of their fleet of legacy applications to determine the return on investment and optimal path forward with respect to the modernization of select digital assets. Importantly, buyers need to understand that modernization is not a teleological process that culminates in the refactoring of monolithic applications into microservices. Legacy application modernization leverages a multitude of approaches and tools and should be understood as a continuous practice or activity that is repeatedly, incrementally performed."

About IDC

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. IDC helps IT professionals, business executives, and the investment community make fact-based decisions on technology purchases and business strategy. More than 1,100 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For 50 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world's leading technology media, research, and events company.

Global Headquarters

5 Speen Street
Framingham, MA 01701
USA
508.872.8200
Twitter: @IDC
idc-community.com
www.idc.com

Copyright Notice