White Paper

Leveraging Multicloud Platforms to Enable Modern Applications

Sponsored by: SHI International Corp.

Sriram Subramanian
June 2020

EXECUTIVE SUMMARY

Organizations pursue digital transformation (DX) to gain a competitive advantage through increased productivity and business agility. Application modernization (AM) is a key step in DX journeys, and enterprise organizations are prioritizing AM in their DX journeys, with agility, competitive advantage, and lower total cost of ownership (TCO) being primary drivers. Enterprises are also facing challenges in their AM initiatives; key inhibitors include the lack of in-house expertise, mismatched expectations, and security concerns.

AM involves modernizing the entire stack — infrastructure, code, and process — with modern infrastructure constructs, such as cloud-centric infrastructure, containers, and serverless infrastructure, providing a strong foundation. Modernizing code involves rewriting some or all of the application to adopt cloud-based services, cloud-native frameworks, and other innovations such as artificial intelligence/machine learning (AI/ML) technologies, Internet of Things (IoT), and edge computing. Process modernization includes adopting agile application life-cycle management processes.

Enterprise applications are critical to the business success of enterprises. In traditional IT environments, the business continuity of applications is ensured through continuous availability and redundancy of underlying infrastructure. Traditional IT processes are designed to avoid any disruption to this continuous availability. Application development practices in such environments also target minimal disruption to application availability and follow traditional software development life-cycle practices such as Waterfall. Enterprises may also be limited to access technological innovations such as AI/ML capabilities without intensive investments. Such dependence of applications on the underlying infrastructure and longer product development cycles limit the ability of the enterprise to innovate faster and to bring products to market more quickly.

Leveraging modern application architectures such as cloud-centric, cloud-native, or microservices-based design enables applications to be more scalable, more fault tolerant, and interoperable. This enables enterprises to deploy the applications on any suitable infrastructure platform. Agile development processes enable application developers to develop and release product features faster. This combination of flexibility in infrastructure platform and development agility enables developers to build faster, thus enabling enterprises to innovate faster.

IDC recommends iterating on a multiphases workload-centric approach, defining the right key performance indicators (KPIs), and leveraging the right partnerships to succeed in AM. IDC also recommends that IT decision makers (ITDMs) prioritize business over technology and leverage

multicloud platforms. Technology buyers should think in terms of solutions versus individual services, leverage right partnerships, and future proof their investments through AM.

This white paper discusses the benefits of AM and best practices to leverage multicloud platforms to enable successful application migration journeys.

SITUATION OVERVIEW

Application Modernization Trends

Application modernization refers to modernizing enterprise applications to leverage modern application architectures, technological innovations, and Agile development practices to gain increased developer productivity and faster time to market.

AM involves modernizing three layers of the entire stack — infrastructure, code, and process (see Figure 1). Infrastructure modernization (IM) refers to modernizing the underlying infrastructure to modern infrastructure constructs such as cloud-centric infrastructure, containerization, disaggregated hardware, edge computing, and accelerated hardware. Modern infrastructure constructs provide scale, flexibility, and an ability to leverage higher-order technology innovations such as AI/ML capabilities at various locations — datacenters, cloud, IoT, or edge. Popular modern application architectures include cloud-native design, 12F applications (a methodology to build scalable and portable software-as-a-service [SaaS] applications on cloud platforms using continuous deployment processes), microservices-based design, and event-driven architecture. This white paper focuses on AM using cloud-native and microservices-based design. Agile development processes, more commonly known as DevOps practices, enable the faster release of applications, thereby allowing enterprises to get products to market faster. DevOps practices also provide a way for developers to consume infrastructure, services, and applications through automation and self-service capabilities. AM enabled by infrastructure modernization and modern development practices is a significant component of DX.

FIGURE 1

Layers of Application Modernization

Process Modernization

- Agile development methodologies
- Automation
- Policy-driven provisioning

Code Modernization

- Refactoring (cloud based, IaaS/ PaaS)
- Re-architecting (cloud native, microservices, serverless)
- Rebuild (cloud native, microservices, serverless, low code)
- Technology innovation adoption (AI/ML)

Infrastructure Modernization

- Cloud-based infrastructure (multi/hybrid cloud)
- Cloud-native infrastructure
- Accelerated/special-purpose hardware

Source: IDC, 2020

An IDC survey (see *Application Modernization Services, 2019,* IDC #US45336616, July 2019) among enterprises shows that nearly 75% of the respondents rated application modernization as a high or top priority. Another IDC survey (see *U.S. DevOps Survey of Large Enterprise Organizations, 2019,* IDC #US45688619, December 2019) reveals that more than 75% of the respondents have already adopted DevOps practices.

As more enterprises are pursuing AM journeys, IDC observes that this is not without challenges. Key challenges include a lack of clarity on what applications to prioritize, a lack of AM expertise, and unrealistic TCO expectations. Though enterprise organizations are adopting DevOps practices, most are reporting less than 20% of their applications using DevOps methodology. IDC research on the maturity of DX (see *IDC MaturityScape Benchmark: Future Enterprise in the United States, 2020,* IDC #US45647618, November 2019) shows that only about 21% of the respondents are at mature stages of their journeys.

Why Application Modernization?

Enterprise applications or line-of-business (LOB) applications, more commonly called legacy applications, are the lifeblood of enterprises. In traditional IT environments, the business continuity of applications is ensured through continuous availability and redundancy of underlying infrastructure. Traditional IT processes, including server upgrades, system maintenance, and software patching, are designed to avoid any disruption to this continuous availability. Application development in such environments also follows traditional software development life-cycle practices such as Waterfall methodologies, thereby requiring longer product release cycles, which are more prone to human error. Enterprises also need to implement any technological innovations in-house in their traditional IT environments for the applications to be able to leverage them. For example, if a legacy application deployed in an enterprise IT environment can be improved through AI/ML capabilities, these capabilities need to be built in the same environment. Such dependence of applications on the underlying infrastructure and longer product development cycles limit the ability of the enterprise to innovate faster and to bring products to market more quickly.

Infrastructure modernization breaks such a dependence on the infrastructure through modern infrastructure constructs, thereby enabling enterprises to deploy the applications on any suitable infrastructure platform. This flexibility enables the application, and hence the enterprises, to leverage any other higher-order services or innovations already available in the infrastructure platform.

AM enables applications to be more scalable, more fault tolerant, and interoperable, thereby making them less dependent on the underlying infrastructure. Agile development processes enable application developers to develop and release product features faster.

Key Drivers

Enterprises cite agility, competitive advantage, and lower TCO as key drivers for AM.

Agility

Infrastructure modernization, through constructs such as cloud-based infrastructure, enables self-service provisioning and infrastructure automation capabilities. These capabilities enable developers to gain operational efficiency and be more productive and agile by self-provisioning resources required for application development and testing. In typical enterprise environments, provisioning a server or a virtual machine (VM) to run tests could take days because of manual, ticketing-based operational processes. With modern infrastructure, developers can self-provision any test environment within minutes.

Another aspect of AM — process modernization — enables Agile development and deployment processes, which empower developers to deliver product features and updates faster.

Thus AM provides application developers with operational efficiency and development agility, which in turn enables them to deliver results faster.

Competitive Advantage

Customers across various industries have been able to stay ahead of their competition by identifying new business use cases and opportunities through AM/DX.

Process modernization, one of the steps of AM, enables business agility. This agility helps developers build products faster, thereby enabling enterprises to innovate faster and bring products to market more quickly. This ability to innovate faster provides enterprises with a competitive advantage over the competition. AM also helps enterprises identify new sources of business opportunities and revenue sources previously not possible.

Lower TCO

Based on conversations with enterprise decision makers, it is evident that modernizing legacy applications enables significant savings in TCO of the IT infrastructure.

AM enables applications to be more resourceful and scale as needed without overprovisioning resources. Popular methods to modernize applications include containerization, refactoring, and re-architecting applications. Containerization refers to packaging applications into one or more containers. Containers provide a runtime environment like that of a virtual machine without the overhead of an entire operating system. Multiple containers can be packed onto a virtual instance or on a bare-metal server, thereby providing opportunities for server consolidation. Using containers instead of full virtual machines also saves on operating system licensing costs because the operating system as a whole is no longer needed to run an application. Because containerization decouples the application from the operating system, it also reduces application dependencies on the operating system and minimizes the need for forced upgrades. With the support of stateful data sets by container orchestration platforms such as Kubernetes, database workloads can also be run on containers, obviating the need for large servers or virtual instances.

Refactoring involves making internal changes in the underlying components to better leverage services available in the cloud. Examples include the adoption of a database as a service, a broader platform as a service, and native cloud-based management tools. Here, the migration integrates the higher-level services offered by the cloud service provider, such as database-as-a-service or serverless functions, toward specific components of the legacy applications.

Re-architecting refers to rewriting legacy applications to more efficient application architectures. The application continues to serve the same business functionality. However, it is now architected to operate under a cloud-native framework (i.e., migrating to containerized/serverless architectures), and architectural changes are external to the cloud service itself but internal to the application. These changes may result in the application leveraging specific cloud-native services on the public cloud.

Application modernization through containerization, refactoring, or re-architecting provides opportunities for resource optimization and hence lower TCO. AM also enables enterprises to consolidate their IT infrastructure resources, thereby helping lower TCO of IT infrastructure.

Key Inhibitors

IDC observes that though more enterprises are prioritizing AM, AM is not without challenges. Enterprises report a lack of clarity and expertise, mismatched expectations, and security concerns as top inhibitors for AM.

Lack of Clarity and Expertise

Most enterprises are still at the early stages of their DX journeys — they are identifying opportunities through experimentation and evaluation. They cite a lack of clarity on what to modernize and a lack of in-house expertise on AM as important inhibitors to their application migration initiatives. This is reflected as decision fatigue and confusion around the "what, when, and how" of AM. These enterprises seek clarity on identifying workloads to modernize, what is the best modernization path for the workloads, and how to prioritize the workloads. They also are perplexed by the complexity of technological innovations and dazzled by marketing hype.

Mismatched Expectations

Not all enterprises that have embarked on AM journeys are satisfied with immediate outcomes. Enterprises report various indicators of dissatisfaction, such as increased TCO, cultural barriers to implementing change, and increased operational complexity.

IDC observes that one of the reasons why enterprises could see an increased TCO, which appears counterintuitive, is unoptimized resource allocation on cloud environments. When enterprises migrate their business applications to the public cloud using "lift and shift" mechanisms, they may overprovision resources on cloud environments. They may not also actively manage resource consumption on the cloud — for example, not turning off services when not in use. As enterprises progress in their adoption maturity, they look at optimizing resources through application refactoring or re-architecting. They also set up necessary automation to ensure the effective management of resources. Until this optimal stage of modern infrastructure consumption, an increased TCO is not unexpected.

As with any change, cultural resistance to adopting Agile development methodologies is natural. IDC observes that with executive sponsorship and a strong mandate from leadership, enterprises can enable organizationwide agile transformations.

In traditional IT environments, IT administrators are used to the consistency and homogeneity across the environment — infrastructure, applications, tools, and processes. IT administrators know how to handle support requests and troubleshoot outages. With the adoption of AM, they need to deal with heterogeneous environments, disparate technologies, varied applications, and new processes in place, which could be daunting.

Security Concerns

Enterprise IT environments are usually fortified with security mechanisms such as firewalls, DMZ, and endpoint security systems. Traditional applications deployed on such environments are also protected by well-established security mechanisms, access control mechanisms, and other protection mechanisms such as encryption. Threat models and attack surface areas of such environments are well understood.

Modern infrastructure constructs extend the boundaries of enterprise IT infrastructure outside the perimeter of traditional datacenters. Currently, a majority of innovation around modern infrastructure constructs is happening around open source projects. Adopting open source projects introduces the possibility of bringing in vulnerabilities from unpatched sources. With increased adoption of distributed repositories such as GitHub, the boundaries of datacenters are blurred. Standardized code review and automated integration testing processes, which are part of Agile development practices, help minimize the introduction of any vulnerabilities in the development process.

Virtual machines are well isolated from each other and the host server. Containers lack such isolation, and threat models on containers are not well understood yet. Managing dependencies across containers is also not trivial, which makes security patching processes complex.

Cloud-based and cloud-native applications deployed on modern infrastructure could have their components across multiple locations. These applications and their components also communicate via APIs through the internet. With the explosive growth of cloud-based and cloud-native applications, security administrators can no longer apply traditional IT security practices and are concerned.

Cloud-Native Applications and Modern Infrastructure Constructs

What Is a Cloud-Native Application?

IDC's public cloud services taxonomy (see *IDC's Worldwide IT Cloud Services Taxonomy, 2019,* IDC #US45714519, December 2019) classifies an infrastructure deployment as cloud infrastructure based on the attributes depicted in Table 1.

TABLE 1

Attributes of Cloud Infrastructure Deployment

Attribute	Remarks
Shared, standard offering	Built for massive scale, automated deployment
Delivered as an all-inclusive service	Pre-integrated and manages/updates all required resources
Elastic scaling	Dynamic, rapid, and fine-grained
Elastic pricing	Tied to resource consumption or number of users
Self-service	Self-service provisioning and administration options
API/published service interface	Programmable access via open/published API

Source: IDC, 2020

IDC's workloads taxonomy (see *IDC's Worldwide Server and Enterprise Storage Systems Workloads Taxonomy, 2020*, IDC #US45936120, March 2020) classifies an application as a cloud-native application if it exhibits attributes such as:

- Auto-scalability
- Auto-redundancy
- Clean separation of state
- Communication through APIs
- Life-cycle management through DevOps processes such as continuous integration/ continuous delivery (CI/CD)

Expanding the aforementioned classifications further, IDC considers an application to be a cloud-native application if it exhibits the attributes discussed in the sections that follow and is being developed through Agile development processes.

Application Attributes

- **Horizontal scalability.** Cloud-native applications exhibit an ability to scale horizontally (scale out) as against traditional applications that scale vertically (scale up). When the traditional application needs to handle more demand, it is scaled up by adding more infrastructure capabilities — such as faster CPUs or more RAM. However, a cloud-native application is scaled up by adding infrastructure resources horizontally — by adding more replicas of the application/service.
- **Resiliency.** Cloud-native applications exhibit resiliency innately without any dependency on the continuous availability of the underlying infrastructure. In cases of any nonavailability of one of the components or the entire application, the cloud-native application can recover automatically. In cases of incomplete recovery, a new instance of the failing service, the component, or the entire application is created to serve the business needs.
- **Statelessness.** An application needs to be as close to being stateless as possible to be fully resilient. In case of the need to store the state of the application, the application exhibits clear demarcation between stateful and stateless components and services of the application.
- **Loosely coupled.** Cloud-native applications exhibit loose coupling among their sub-components and services, as opposed to traditional applications that tend to be strongly coupled. For example, in a typical three-tier web application, each tier is strongly connected to others in terms of the code base, runtime, deployment configuration, execution environment, and more. Updating or fixing a bug in the presentation layer requires the entire application to be built, tested, and deployed. One must exercise extreme caution to ensure no data loss and business continuity. Such a tight coupling between tiers (components) slows the possibility of rapid innovation. However, it is easy to update a cloud-native application because its components and services are loosely coupled and independently managed.
- **Packageability.** Cloud-native applications exhibit an ability to be packaged into a collection of independent services that can be individually deployed and managed independently of other services. This ability of cloud-native applications enables horizontal scaling of the application rapidly. Containers provide the right level of infrastructure abstraction to package these services, thereby providing the best resource optimization. Such packaging of individual components also enables Agile development and deployment of services.
- **API-based interaction.** Components and services of a cloud-native application use APIs to communicate with each other. Cloud-native applications also support API endpoints, usually implemented as REST APIs, for external interaction.

Development and Deployment Processes

- **Agile development methodologies.** Cloud-native applications are developed and managed through Agile development methodologies — such as DevOps practices, with each component or service being developed, managed, and deployed independently. A single cloud-native application could have multiple CI/CD pipelines for managing constituent services. While the attributes of cloud-native applications such as packageability and loose coupling enable the cloud-native application to be rapidly developed, Agile development and management capabilities make rapid development possible.
- **Extreme automation.** Cloud-native applications rely on extreme automation for management. Because they have various independent moving parts (services and components), they rely upon automation for orchestration, registry, health checks, and other aspects of management.
- **Policy driven.** Because cloud-native applications are managed through automation with less or no minimal human intervention, they rely upon policy-driven resource provisioning to ensure efficient and effective resource consumption. Policy-driven resource provisioning also enables developer- or code-led ways to provision and consume infrastructure resources.

The term *cloud native* refers to a generic set of applications that exhibit the aforementioned attributes and are managed using the aforementioned methodologies. It is also important to note that the term *cloud* refers to the ability to scale like cloud-based infrastructure, not necessarily a deployment location. A cloud-native application can be deployed on a public cloud platform or on-premises environments. A cloud-native application can be built using virtual machines or containers, with containers being the most adopted infrastructure abstraction to enable cloud-native applications. Kubernetes, an open source container orchestration platform, has proven to be the most widely adopted orchestration platform to deploy cloud-native applications.

Multicloud Platforms as an Enabler

Modern infrastructure constructs such as cloud-based services, containers, and serverless abstractions provide an ideal platform for AM. Multiple cloud service providers support these modern infrastructure constructs to enable all paths to AM, thereby proving to be a strong foundation for the same.

Cloud-native infrastructure refers to infrastructure abstractions that enable cloud-native applications. As discussed previously, while cloud-native applications can be implemented as virtual machines or containers, containers on Kubernetes prove to be the most adopted mechanism to implement cloud-native applications.

Kubernetes supports orchestrating containers on both bare-metal servers and virtual machines across on-premises and public cloud platforms. This ability to support container orchestration across heterogeneous environments enables cloud-native applications to be portable across on-premises and public cloud environments.

Serverless infrastructure abstractions enable specific executions (called functions) to run without needing any computing resources (bare-metal servers, virtual servers, or containers). The abstractions provide the ability to run these executions upon some triggers, such as events.

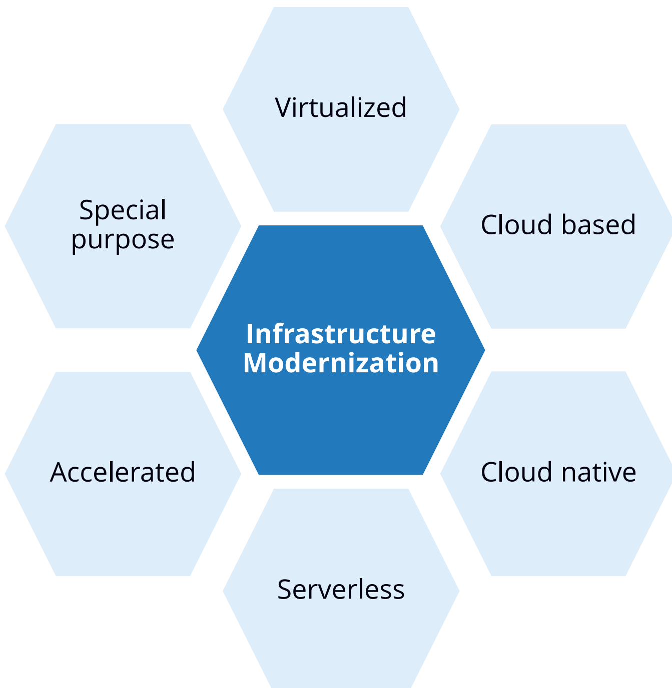
Low-code and no-code paradigms refer to the ability to implement business logic (such as sending an email confirmation) without the need to write any code. One can build a complete business application by simply assembling low-code or no-code blocks, just like how one would build using Lego blocks.

The past few years have also witnessed tremendous progress in innovations such as AI/ML technologies, enabled by easier availability of accelerated hardware, the democratization of AI/ML technologies, and innovations in open source projects. Public cloud service providers also offer AI/ML capabilities as services, which enables application developers without data science expertise to infuse their applications with AI/ML capabilities. Cognitive capabilities such as video recognition, natural language processing, image recognition, and facial recognition enable innovative use cases, which may not be possible without such capabilities.

By supporting various infrastructure constructs (see Figure 2), cloud service platforms provide an ideal platform for AM. Enterprises can leverage multiple cloud service platform services and capabilities across multiple cloud service providers based on their needs. For example, enterprises can standardize cloud-native applications on containers running on Kubernetes clusters, database capabilities consumed as services, end user–facing capabilities implemented as functions, and certain stateful legacy applications rehosted to VMs on cloud platforms.

FIGURE 2

Modern Infrastructure Constructs



Source: IDC, 2020

Best Practices for Application Modernization

Iterate on a Multiphased Application Modernization Journey

IDC's MaturityScape Benchmark provides a repeatable framework to evaluate the maturity of a journey (say, DX) or technology adoption (say, artificial intelligence) by an organization. As organizations advance in their journey or adoption, this framework helps them evaluate their progress and proceed further.

IDC recommends considering the maturity levels as guidance and pursuing a multiphased approach to move forward in the AM journey. An enterprise should begin with infrastructure modernization, followed by code modernization and, finally, process modernization. IDC also recommends iterating on these phases with overall improvements between iterations.

Infrastructure Modernization

During this phase, IDC recommends that enterprises build a strong foundation on a public cloud platform by migrating their legacy workloads to a public cloud platform. This phase is also the right time to make an inventory of business applications and IT infrastructure, which would also help organizations identify the priority of workloads to be modernized.

This type of migration of workloads to the public cloud enables organizations to develop a footprint on the public cloud rapidly. During this phase, organizations can also experiment with containerizing smaller applications by moving them to containers (instead of virtual machines) without significant changes to the application. This type of containerization is equivalent to a lift and shift to VMs. With this footprint on the cloud as the foundation, enterprises can explore the next phase of AM — code modernization. During this phase, IDC also recommends experimenting with and evaluating platforms that enable interoperability, such as Kubernetes.

Code Modernization

During code modernization, IDC recommends that enterprises refactor (to leverage cloud-based services) or re-architect workloads (to leverage cloud-native frameworks).

Refactoring legacy workloads provides an opportunity to adopt cloud-based services such as database as a service or managed databases as data back ends for applications. Such refactoring of applications also helps enterprises ramp down capex investments and operational overhead.

Re-architecting workloads provides an opportunity to rewrite the applications using cloud-native/microservices-based architectures. IDC recommends leveraging platforms such as Kubernetes to enable such workloads. Re-architecting workloads also offers an opportunity to rewrite and rebuild the applications using serverless or low-code/no-code paradigms.

Process Modernization

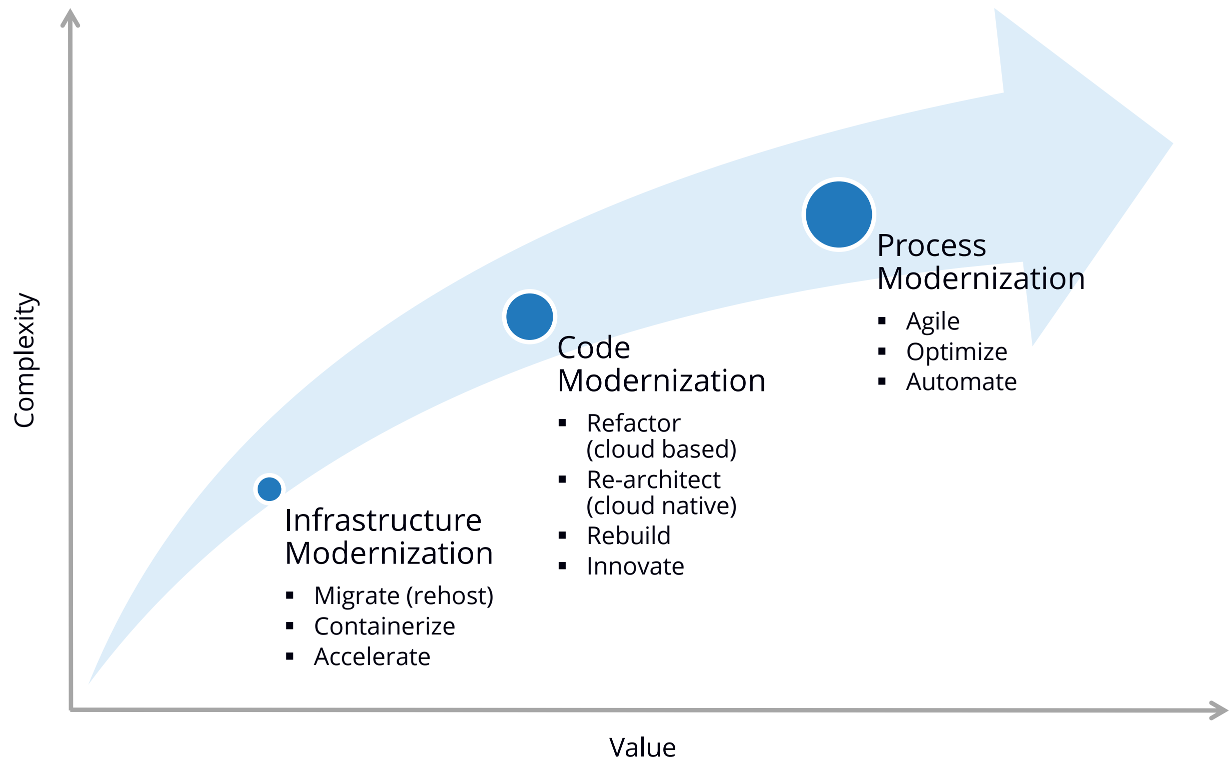
Process modernization refers to adopting Agile development practices such as DevOps methodologies. During this phase, IDC recommends that organizations transform their culture to adapt to these practices and succeed in leveraging DevOps methodologies. IDC advises deploying the right set of tools (such as source code management, issue tracking, testing, automation, and deployment) and processes (such as CI/CD) in order to adopt DevOps methodologies. Enterprises also need to invest in training and education opportunities on Agile development methodologies, tools, and processes.

Cultural transformations and organizational changes such as DevOps transformations are rarely successful without executive sponsorship. IDC recommends securing support from the leadership before expecting enterprisewide transformations.

As an organization goes through the three phases of AM, its level of AM maturity increases and the value actualized also grows. It is important to note that with progress in AM maturity, the complexity of managing these applications also increases. However, with mature automation and processes, the organization is better positioned to handle this increasing complexity. IDC also recommends iterating on these phases with improvements across these phases for every iteration. Figure 3 summarizes the concept.

FIGURE 3

Phases of Application Modernization



Source: IDC, 2020

Take a Workload-Centric Approach

During each phase of AM, IDC recommends taking a workload-centric approach to determine what is the right path to modernization. Paths to AM include rehosting, refactoring, re-architecting, or replacing with equivalent SaaS offerings. Because replacing is more dependent on the availability of equivalent SaaS offerings than on the nature of the application, this section discusses only the other paths.

One can view the selection of an appropriate path for AM as a function of how modular the application is versus how stateful the application is. Highly modular workloads are loosely coupled and hence can be broken down into smaller components and easily re-architected into microservices-based architecture. Traditional monolithic applications are less modular and tightly coupled. Statefulness refers to the need for the application to store its state — the more stateful an application is, the more its dependence on persistence storage back end, and hence the more challenging it is to adopt cloud-native infrastructure.

Rehosting involves moving applications to virtual instances on public cloud infrastructure with no or minimal code changes. The application continues to behave the same way as it did on on-premises infrastructure. The application would need support from the underlying infrastructure for availability and scale. The application also continues to be managed in the same ways as it was on on-premises infrastructure. Rehosting is recommended for applications with tight coupling between components.

Containerization refers to rehosting applications to containers instead of virtual instances. While containerization provides additional advantages such as easier deployment and life-cycle maintenance, containerization does not involve significant code changes to the application. Containerization is recommended for smaller applications that are candidates for rehosting.

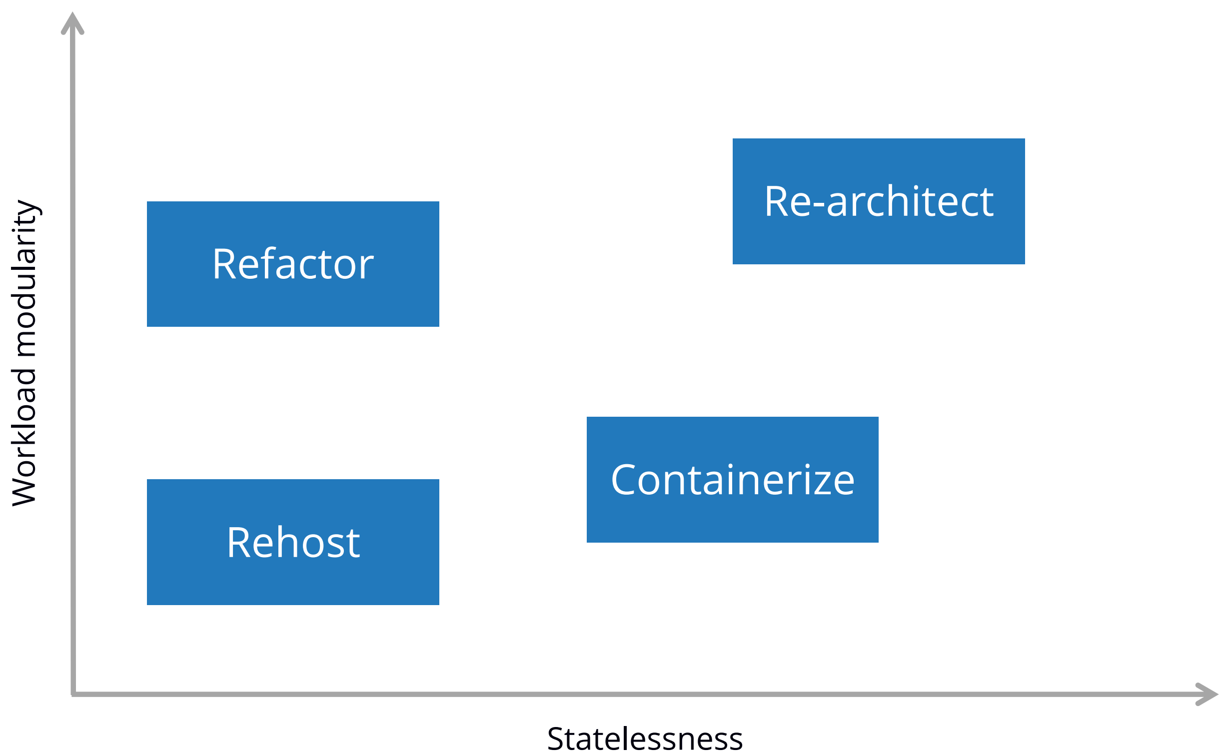
Refactoring refers to changing the internals of the application to leverage cloud-based services without significant architectural changes. Refactoring is recommended for stateful workloads with looser coupling.

Re-architecting involves rewriting the application completely to leverage cloud-native technologies. Re-architecting is recommended for less stateful workloads with even more loose coupling of components, event-driven applications, and net-new, modern applications that are written from scratch.

Figure 4 summarizes the different approaches to workload-centric application transformation.

FIGURE 4

Workload-Centric Approach



Source: IDC, 2020

Define and Measure Success

It is essential to define the right KPIs to measure the success of AM. While the maturity of the organization is a good indicator, specific KPIs help organizations measure success. IDC recommends KPIs to measure business, financial, and operational success. It is critical to define unambiguous, actionable, and timely KPIs that can be measured periodically.

This section discusses representational KPIs that the reader can adopt to evaluate the progress of the AM journey.

Recommended KPIs

Business

Business KPIs enable measurement of business success, such as expanding business opportunities to three additional areas, increasing revenue by 10%, identifying two new sources of revenue, and improving customer satisfaction by 15%.

Financial

Financial KPIs enable measurement of financial success, such as decreasing capex by 5%, increasing DX investments by 10%, and increasing profitable customers by 50%.

Operational

Operational KPIs enable measurement of operational success, such as decreasing the spend on system outages by 10% and increasing product release frequency by 300%.

Measuring KPIs

IDC recommends that organizations use a performance scorecard to measure success against identified KPIs. For each KPI, organizations should evaluate whether their performance is leading or lagging the planned measures. In the case of leading indicators, they should continue their investments and plan and adjust behavior as required. In the case of lagging indicators, organizations should assess why the performance was off and redirect resources accordingly.

IDC also recommends planning for the long term, with different priorities — opportunistic in the short term (one year), incremental in midterm (two to three years), and transformational in the long term (more than three years).

Leverage the Right Partnerships

One of the critical challenges inhibiting AM, as cited by enterprise customers, is the lack of in-house talent. IDC recommends that organizations partner with the right solution provider to mitigate this situation. Solution providers can help organizations identify the right cloud platforms, tools, and technologies to employ; determine the appropriate path for AM; and help during every phase of the modernization journey. Solution providers can also help organizations expand on in-house expertise through training and knowledge transfer.

Solution providers are also ideally suited to integrate offerings and services from cloud services providers into complete solutions that can accelerate AM.

For example, consider the case of an enterprise looking to modernize a Windows .NET application with SQL Server as the back end. A solution provider with expertise on cloud platforms, middleware, containers, and databases can help the enterprise succeed in modernizing this application by integrating various technologies across multiple platforms, such as the following:

- Migrate applications to virtual servers and virtual storage on one or more cloud platforms.
- Refactor application back end to leverage databases on cloud platforms or cloud-based database services as data back end.
- Re-architect application components to microservices that can be run as containers on on-premises or cloud platforms.
- Establish Agile development practices using CI/CD platforms.
- Leverage cloud-based middleware services.

The provider can also help test, deploy, and manage this application on the cloud platform of choice.

Enterprises should ideally partner with solution providers that have expertise on multiple cloud platforms, cloud-native technologies, and domain-specific knowledge.

FUTURE OUTLOOK

IDC's Take

IDC observes that the global spend on public cloud services has grown significantly in the past few years, and more enterprises are increasingly adopting public cloud platforms. Enterprises are also moving their business-critical and mission-critical workloads to cloud platforms. IDC also observes that while more enterprises are prioritizing AM initiatives, most of the enterprises are still at the early stages of AM. A recent study of enterprise workloads on public cloud infrastructure (see *Enterprise Workloads on Public Cloud Infrastructure,* IDC #US45959320, February 2020) shows that most enterprise workloads are migrated to the public cloud using lift and shift mechanisms, indicating ample opportunities for refactoring or re-architecting.

CHALLENGES/OPPORTUNITIES

Challenges

The lack of in-house expertise, unrealistic expectations, and security concerns are significant inhibitors to AM. IDC recommends that enterprises pursue a workload-centric, multiphased approach to AM with a multicloud environment as the enabling foundation.

The combination of the adoption trends among enterprise organizations and the state of adoption maturity provides credible opportunities to cloud service providers and solution providers.

Opportunities

Opportunities for Cloud Service Providers

While the adoption of public cloud service platforms among enterprises is proliferating, less than 20% of enterprise workloads (based on conservative estimates) have been migrated to the cloud. A vast majority of these migrations are based on lift and shift approaches and hence do not leverage the operational efficiency and cost benefits that the public cloud platform can enable.

This trove of legacy applications within on-premises datacenters and the mandate to modernize applications provide tremendous opportunities for cloud service providers. Such opportunities include, but are not limited to, infrastructure-as-a-service [IaaS] and platform-as-a-service [PaaS] revenue, managed services revenue, and professional services/consulting opportunities. It is not surprising that almost all major cloud service providers offer migration services directly or through their technology partners. Apart from contributing to the top-line growth of cloud service providers, these opportunities also enable end users to succeed in their application migration and, hence, DX journeys.

Opportunities for Solution Providers

While cloud service providers are well positioned to gain from growing application migration initiatives among enterprises, they also limit enterprise customers to a single cloud service provider. IDC also observes that enterprises are repatriating specific workloads to on-premises infrastructure, citing resource consolidation, TCO, and data security concerns as key drivers for such repatriation. Solution providers are uniquely positioned to address these limitations.

Solution providers with expertise on multiple cloud service platforms can help organizations select the right cloud platforms for their workloads, including on-premises environments. They can integrate services across multiple cloud service providers as needed. Solution providers also have deep vertical-specific domain expertise, which can accelerate modernization of vertical-specific applications.

CONCLUSION

Recommendations for ITDMs

IDC recommends that ITDMs consider the strategies discussed in the sections that follow to succeed in their AM journeys.

Business First, Technology Next

One of the common patterns among enterprises that cite mismatched expectations from AM is adopting new technology or innovation without validating whether that is the right choice for their use case. Such a technology-first approach could be counterproductive, resulting in employee dissatisfaction, increased complexity, potential loss of revenue, and unmet TCO expectations. Enterprises are also most likely to spend more effort and make greater investments in trying to reverse poor technology choices.

IDC recommends approaching AM from a workload-centric point of view to select the right paths, tools, and technologies for the workload. If a business workload does not need to be reimplemented as serverless functions or be AI enabled, then there is no need to take any other approach with that workload. Enterprises should also adhere to recommended best practices on selecting KPIs. Such an approach prioritizes business needs over technology hype, thereby enabling ITDMs to make the right decisions.

Leverage Multicloud Platforms

Multicloud platforms enable successful AM initiatives by providing flexibility and choice. By selecting appropriate technologies, tools, or services from multiple service providers that best fit the business need, enterprises are not limited by what is provided by a single cloud service provider and can leverage best-of-the-breed capabilities. Multicloud platforms also offer enterprise organizations an advantage in pricing negotiations with cloud service providers.

IDC recommends enterprises take a multiphased approach to migrate workloads to public cloud (see *Five Phases of Cloud Migration: A Workload-Centric Discussion on Planning Through the Journey,* IDC #US45328519, July 2019). IDC further recommends expanding this strategy across multiple cloud service providers to leverage appropriate capabilities from these platforms to succeed in AM initiatives. It is important to note that adopting multicloud platforms is not mandatory for AM, but enterprises should consider leveraging multicloud platforms without being restricted to a single cloud service provider.

Pursue as a Marathon, not a Sprint

Initiatives such as DX and AM are impactful across the entire organization and hence bound to take a long time to show the full effect. An enterprise should consider such initiatives as marathon races, where consistent performance over a longer duration is needed, versus sprint races, where one needs to perform at peak performance for a short period.

When enterprise applications are migrated to a public cloud platform using lift and shift mechanisms, enterprises may not provision resources on cloud optimally. Pay-as-you-go pricing models offered by public cloud service providers are also not cost effective for workloads that need to be kept on 24 x 7. Because of these reasons, it is entirely expected that some enterprises may find that cloud spend is more expensive during the initial stages of AM. It is thus important for organizations to have this long-term view rather than focus on immediate gains or losses. In this long-term view, IDC recommends having different focuses across horizons — short term (opportunistic), midterm (incremental), and long term (transformational).

Define the right success metrics/KPIs around business, financial, and operational goals, and evaluate organizational progress against set KPIs periodically. Enterprises should act based on leading or lagging indicators and adjust priorities if needed, just like how a marathon runner course-corrects during the race.

Recommendations for Technology Buyers

IDC recommends that technology buyers consider the strategies discussed in the sections that follow to succeed in their AM journeys.

Leverage the Right Partnerships

For enterprises that lack in-house expertise on cloud-native technologies, having the right partnerships could be the most significant difference between success and failure in their AM journey. Enterprises have the choice of partnering with cloud service providers, professional service providers, managed service providers, or solution partners. IDC recommends leveraging the right partnerships to mitigate the lack of in-house expertise and increase the chance for successful application migration and DX journeys.

Buy Complete Solutions, Not Individual Offerings

As discussed previously, AM consists of infrastructure, code, and process modernizations. While public cloud platforms provide various capabilities that enable infrastructure and code modernization, successful modernizations need more than just using disparate services. Cloud services can be compared to Lego blocks: Blocks by themselves do not make any sense, but when assembled, they can make meaningful and intricate shapes.

Pricing models offered by cloud service providers are granular and services centric. While such models are cost effective when used optimally, they also introduce unpredictability and complexity in TCO calculations. On the other hand, pricing for solution offerings, though more expensive, are predictable and easier to understand.

IDC recommends that technology buyers make purchasing decisions around complete solutions rather than piecemeal services to ensure predictability and completeness.

Future Proof Your Investments

Infrastructure innovations and application development paradigms and architectures go through mutually influencing cycles of innovations. By investing in the right tools, technologies, processes, and culture, an enterprise organization is well positioned to leverage the latest innovations and to adopt upcoming innovations. IDC recommends that enterprises modernize their business applications through infrastructure, code, and process modernization to be ready for future innovations.

MESSAGE FROM THE SPONSOR

Application Modernization by SHI International Corp.

SHI is an $11 billion global provider of technology solutions, products, and services. With a dedicated customer-centric approach, SHI helps customers select, deploy, and manage the technologies they need to meet their business objectives.

SHI has been driving technology innovation since 1989 and supports organizations in the strategy, design, deployment and management of hybrid infrastructure and modern applications.

Today, over 15,000 customers rely on SHI as a trusted advisor to enhance their applications and infrastructure. For more information on how application modernization can help your business, visit https://shi.com/application-workloads.

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