

Considering Leaving Legacy IBM Platforms? Beware, as Cost Savings May Disappoint, While Risking Quality

Published 21 March 2019 - ID G00383838 - 16 min read

By Analysts [Thomas Klinect](#), [Mike Chuba](#)

Initiatives: [Application and Product Portfolio Governance](#) and [1 more](#)

Legacy platforms may seem old, outdated and due for replacement. Yet IBM and other vendors are continually integrating open-source tools to appeal to more developers while updating the hardware. Application leaders should reassess the capabilities and quality of these platforms before leaving them.

Overview

Key Challenges

- Legacy platforms can be a convenient scapegoat for corporate issues with technology, increasing pressure on application leaders to decommission them without careful analysis.
- Organizations frequently consider modernizing their platforms as an IT issue rather than a business decision. IT-driven modernization efforts often fail, increasing both cost and complexity.
- Many business and IT leaders believe traditional systems are expensive, yet after calculating the total cost of modernizing applications to new platforms, revelations show there is very little benefit in actually making the move.

Recommendations

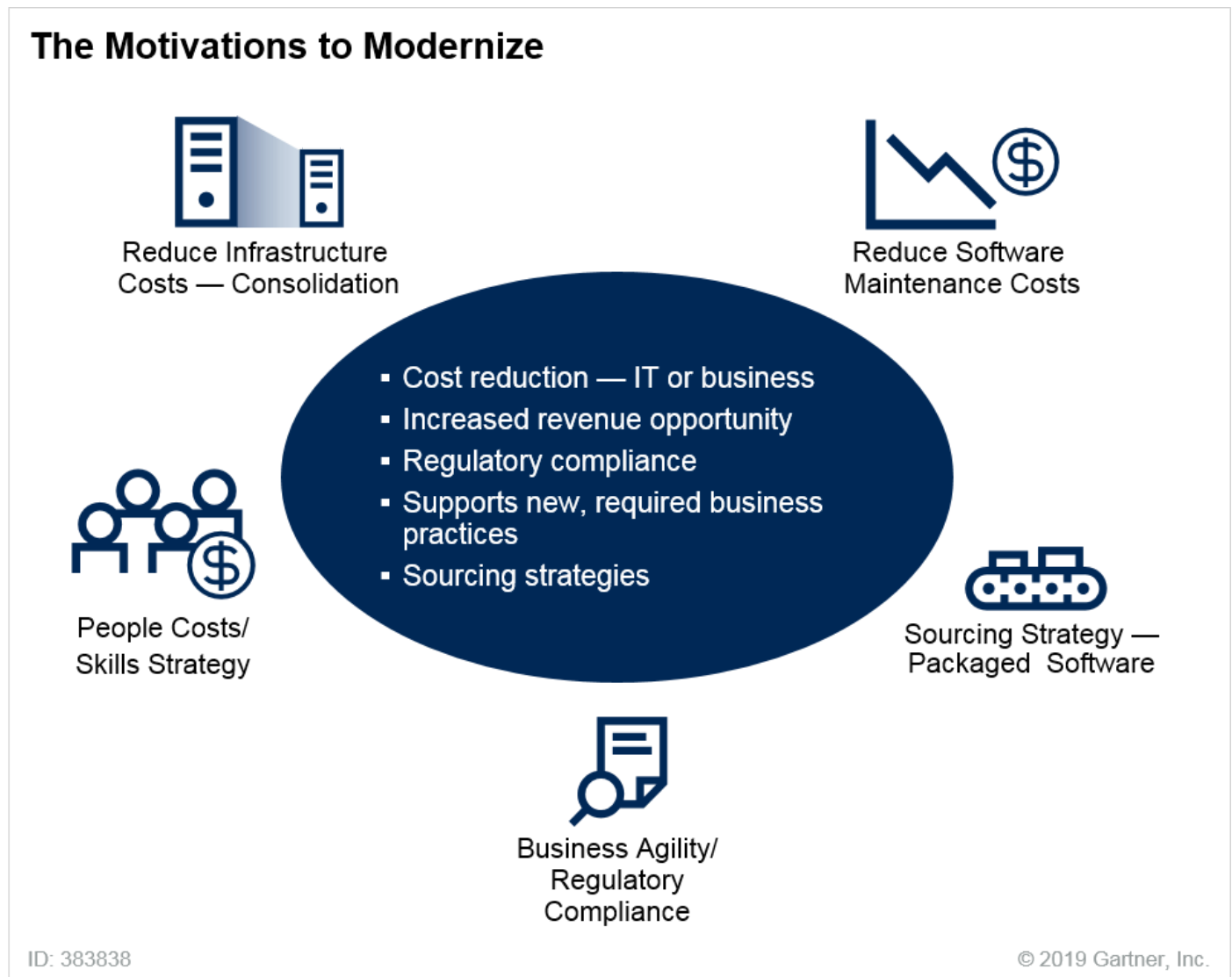
Application leaders considering application modernization while governing application and product portfolios should:

- Remove emotion from the decision-making process by conducting a business-led, IT supported audit of the entire ecosystem, in order to understand the business benefits of a platform change.
- Analyze existing technology and capabilities as well as the quality of traditional platforms by documenting the gap between business requirements and the platforms supporting them.
- Uncover the total cost of ownership of transitioning from the existing platform to another by performing a fact-based analysis of current expenditures, as well as contrasting the target platform expenses, being sure to include total transformation costs in the budget.

Introduction

Pressure for application leaders to jettison current corporate systems and modernize to newer platforms is growing. Emotionally charged terms like “fragile,” “legacy” and “technical debt” lead to the assumption that older so-called legacy platforms and software must be replaced. (Gartner prefers to use the term “traditional” to describe these platforms.) Modernization programs change technology, platforms and ecosystems by replacing technical debt with current technical capabilities.

Figure 1. Modernization Motivations



Source: Gartner (March 2019)

Modernizing platforms can be expensive and high-risk, and is not guaranteed to improve support for critical strategic applications. On average, the cost of modernizing systems of record routinely swells into the tens of millions of dollars. By the end of the program, companies risk ending up with more complexity, more technical debt and more support issues than when they started.

Should a company modernize?

*The answer depends on each organization's needs, the capabilities of its core compute platforms, and its tolerance for risk. Analyze the intersection of technology and business needs, by working with IT in conjunction with business stakeholders to determine whether to refresh or replace current platforms. Sometimes modernizing 40-year-old software **is** the right answer. Yet you need to perform a proper analysis to make the correct decision.*

Analysis

Remove Emotion From the Decision-Making Process

Replacing existing systems because people perceive them to be old can be a costly mistake. Instead of making a monumental decision based on the misguided perception that current platforms are too costly, measuring the total platform benefit to business is a better test. Additionally, you must consider the **total** cost of platform abandonment.

CIOs and other IT leaders involved in the decision to abandon the platform may not have the most current information about the capabilities of the traditional platform. As retirement of CIOs and other senior leaders takes place, the leadership team loses perception and understanding of the reasons past leaders chose the platform. Not having “grown up” with the platform, new leaders often lack a deep understanding and knowledge of its capabilities. There may be misperceptions or outdated information — for example, its ability to support open systems. The bright and shiny marketing rhetoric used to sell new systems makes traditional platforms seem inferior.

The primary driver for the replacement of traditional platforms should always be business. When business is incapable of keeping up with the pace of the marketplace, or the technology in business prevents the expansion of business capabilities, only then should you construct a business case to abandon a platform. Even then, total replacement of the traditional platform is rarely warranted.

Replacing perceptions with facts is crucial to make the best decision for the enterprise. Work with colleagues across the organization to determine if the platform is harming competitiveness. If business colleagues and your root cause analysis cannot point to a single platform as the reason for their noncompetitive capabilities, there is no business case to abandon that platform.

Begin this analysis by asking nonleading questions to determine whether the platform is causing problems. Obtain complete answers to questions — such as the following — before even beginning to consider replacing functional platforms:

- What is business unable to do to stay competitive/relevant?
- How many change requests to your business application are still outstanding?
- When looking at the steps needed to perform a business workflow, which steps require more time than you think they should?

- How often does one of your main workflows fail (in part or whole)?

Understanding the issues which persist in the business space is critical in gaining business value from modernization. Statements such as “it doesn’t work fast enough” provide no value to any organization in a modernization program. Other specific questions to ask include:

- Does data movement cause a bottleneck?
- Does a lack of input/output (I/O) bandwidth cause an application to perform poorly?
- Is the database being overwhelmed?
- Is the display used by customers or business employees inflexible or hard to use?

It is important to limit questions that elicit information about how the platform affects the processes the business uses to generate revenue for the business. This means excluding any questions related to IT. Including IT-related questions allows perceptions based on the age of the system to wrongly influence the decision on whether to replace it.

The objective is to determine whether the traditional platform is helping the business or hindering it from meeting its goals.

Analyze Traditional Platforms by Documenting the Gap Between Business Requirements and Their Supporting Platforms

Any platform is only as good as the value it delivers to business. Using business requirements to determine which platform is adequate for each application, application leaders can develop an application platform strategy that best suits those business needs. Platform choices affect the business only insofar as they affect the business-visible aspects of how well the application achieves its mission, with what quality and at what cost.

Some companies that abandoned traditional platforms have come to regret the transition later on. Figure 2 illustrates some of the priorities you should take into account before making such a drastic move. Performing platform due diligence ensures that the decision to move is not predicated on hearsay and popular lore. Whether you choose to abandon your core platform or not, there are ways to lower your risk and assess your true cost savings.

Figure 2. Set Modernization Priorities

Set Your Modernization Priorities

Which problem do you want to fix first?

Cost	✓
Agility	✓

Do you have existing Linux/Windows infrastructure you want to leverage?

Existing hardware	✓
Existing skills	✓
Operational robustness	✓

If you have an IBM mainframe, have you already maximized cost savings opportunities on that platform?

Specialty engines	✓
Software pricing	✓
Replace expensive products with lower-priced alternatives	✓
Rationalized system software	✓
Rationalized application portfolio	✓

ID: 383838

© 2019 Gartner, Inc.

Source: Gartner (March 2019)

Some IT leaders believe they are moving to a less expensive hardware environment, with the mistaken belief that one can overpower a tool problem with hardware. The historically low average utilization level of distributed servers has called into question the efficiency of those platforms when compared with the mainframe environment. Even with virtualization, the utilization of these platforms routinely fall well below the continuous utilization rates of traditional systems. It is not all about clock cycles. Input/output is the typical bottleneck for applications. Here, these traditional systems shine.

In a similar vein, virtualization products from VMware, Microsoft and others have brought significant benefit to users. However, the mainframe's virtualization capabilities remain the gold standard for those that have extremely advanced levels of computing.

After interviewing or surveying business leaders as described in the previous section of this research, the next step is to use the answers to the questions to determine if abandoning the traditional platform is the right move. To choose a clear course of action, perform a complete analysis based on these statements from the business – not on the age of the platform or software. Traditional platforms offer many tools for analyzing workloads to inform your decision.

Considering the results of such analyses from many organizations, the issues businesses most commonly confront are in the customer experience and workflow, rather than core transaction processing. Today, these lightweight components are easily extracted from the monolithic applications. This allows application leaders to move the lightweight parts of a business workflow to less expensive commodity platforms. This “balancing the enterprise” method allows application leaders to allocate more of the system resources of the mainframe or IBM i to the actual computational and I/O intensive functions where the mainframe excels.

New Tools Remove the Gap in Delivering Business Value Between Platforms

When comparing various possible modernization scenarios, it's important to consider – along with “status quo” versus “leave the platform” – other options for modernizing currently in place. The range of possibilities for this is quite broad, but the most important advances exist in development tools, testing and distribution into production. If your initial comparison raises significant issues with the status quo, but also raises concerns about the viability of leaving the platform, then analyze these areas more thoroughly to build out your full range of options to inform your action plan.

Development Tools

Many application leaders have lamented on the poor quality and quantity of tools for traditional platforms, such as the mainframe or IBM i. While these platforms have a much smaller ecosystem of independent software vendors (ISVs) supporting them than distributed platforms, the introduction of open source to the mainframe is changing this.

Now developers, testers, and infrastructure and operations (I&O) staff have the capability to utilize the same tools which exist in the distributed world. Rocket Software, CA Technologies and IBM are supporting the Open Mainframe Project Zowe, which is making adapting of open-source tools much easier.

From application development software, to complex DevOps orchestration engines, these traditional platforms are enjoying a resurgence of relevance in the data center that is making them accessible to all developers and testers (see [“Extend DevOps Across the Enterprise by Embracing Open-Source Toolchains for the Mainframe”](#)). This change significantly reduces the issue of

limited and old-fashioned development tools that previously contributed to the impetus to leave traditional platforms.

Software Testing

Moving from waterfall to agile methodologies allow development teams to generate more value more quickly. As the pace of software development heightens, the pace of testing must also increase. With new open-source tools and agile techniques, traditional testing organizations possess the capability to test more code more often. This concept breaks the long-held idiom of “production is test” on these systems of record where what testing occurred was mostly manual in the final weeks before launch. Since the ported open-source software (OSS) testing tools are now oblivious to the underlying hardware, their use in a multiplatform environment blurs the line as to what was once a much-regimented process.

The creation of test routines for all phases of testing is now possible utilizing new frameworks and tools brought on to the traditional platforms by the open-source community. Offloading test to commodity x86 platforms, utilizing the IBM’s Z Development and Test (ZD&T) environment, allows the continual testing of software. The automation that open-source test tools provide allows developers to produce applications with far fewer technical issues and security vulnerabilities. This improvement leads to more stable applications in production. As software resilience increases, organizations find they spend more time on innovation and less time and money supporting the production environment.

Software Distribution Into Production

For 40 years, application and business leaders searched for an easy method to deploy applications into production. Traditional tooling is rigid and vendor-specific. Update periods must be implemented to deploy code into production, which causes problems for support personnel and deploying critical bug fixes.

Utilizing software distribution software familiar to the open-source community, like Bamboo, allows mainframe and IBM i platform owners to easily push packages into production. Now when business requests new features, teams can deliver these features in smaller releases, more frequently.

The monitoring of software and environments on mainframe and IBM i platforms far exceeds anything available today on commodity x86 platforms. The existing capability to see into every nook and cranny of these compute platforms still remains, and is only enhanced by open-source software’s capability to log and display that information. Through introducing this new software, I&O leaders now possess the capability to visualize the platform in ways not imaginable five years ago.

Application leaders are developing software at an ever-increasing pace. Test teams now provide full coverage testing support for this new deluge of applications. Along with the ability to monitor

every step of the applications' journey, at such a low level, this provides an advantage unparalleled in the distributed world.

The capability to monitor all of the applications and display that information on dashboards across the entire ecosystem makes these traditional platforms more than relevant in business — again. Business leaders — still unaware of which platforms are running their applications — can obtain the capability to deliver new features and functions to their customers at a pace which rivals the commodity x86 platforms. All the while providing “five-nines” reliability and unparalleled security.

Determine the Total Cost When Considering Moving Off a Legacy Platform

Determining the total cost of ownership of a traditional platform requires taking many items into account. Some of the items are “nonfunctional,” such as the governance changes an organization must adopt. The acquisition of hardware is a capital expenditure (capex) expense, which turns into an operating expenditure (opex) expense, as the continual upgrading of hardware is required. Therefore, board approval (or senior executive approval) to acquire hardware is required.

This makes a fact-based business case an absolute requirement (see [“Building a Multiplatform Application Modernization Business Case”](#)). Many business cases overlook the actual cost of transitioning a business to the desired target state. Therefore, the true cost of ownership is never exposed for analysis; otherwise, the ROI ratios would not look as good as they do without this information.

Some items that you must explore when considering abandoning a platform are:

- Cost of new platform (cloud subscription, new applications, among others)
- Modernizing existing applications to new platform (long ROI)
 - Cost, including change orders
 - Risk
 - SLAs
 - Cost to create change data capture (CDC) — required for parallel run
 - Cost of maintaining two systems for “N” months
- Training/retraining costs
- New application development and maintenance (ADM) processes
- Security exposure
- The accelerated replacement cost for commodity platforms

- Cost of decommissioning the current platform
- Interruption to business

While some of these items already exist in the ecosystem, absorbing the applications from the abandoned platform, rarely are all these costs included in the final calculation. Therefore, despite the fact some of the items listed previously already exist, the cost of their increased utilization must be factored into the overall equation (including the cost and complexity jump, as different modernization methods are employed for each application). Organizations often find they have at least three modernization methodologies in-flight at the same time:

1. Rearchitect
2. Replatform
3. Retirement

Each methodology has its own distinct and separate costs. Just replatforming an application brings the requirement for continual language support and maintenance. Rearchitecting an application excludes the existing support staff from ADM as the code is no longer in its former language, nor does it even look the same.

Then there is the process of moving an application from a monolithic state to a mini/macro/micro service component. Moving from a stateful to stateless form carries great risk which needs to be managed alongside other moving parts of the modernization. “Greenfield” development efforts are the most expensive modernization methodology. Even if the application leader extracts the current business rules from the current platform, building an application around that logic is fraught with perils. The testing phase of the greenfield methodology takes longer than other methodologies since the code is brand-new and hand-written, which proves more expensive than other modernization methodologies.

Even with the latest agile and DevOps practices, moving application code from one platform to another is extremely risky. The continual failure to transform applications from these stalwarts’ systems has created a complexity so difficult to understand – and even more difficult to manage – that most organizations no longer aim to abandon traditional platforms as part of their modernization program. At most, they work only gradually toward this objective, in a series of smaller, lower-risk steps (see [“Use Continuous Modernization to Build Digital Platforms From Legacy Applications”](#)).

With new tools entering the analysis space, application leaders can finally understand how business processes utilize their assets. Application modernization programs can now move forward and modernize both on and off platforms the components best suited for the platform.

Then there is the reliability of the platform. For applications which need five-nines (99.999%) uptime, these traditional platforms are still the best location to run them. However, running web servers and API gateways is better served on the economical commodity platform found in most data centers, or in the cloud. For on-premises use, uptime can be augmented through redundant systems behind a load balancer to gain a 99.95% uptime. Cloud platforms can also be configured for high availability, but at added cost, and only for applications correctly written to leverage the mechanisms provided to enable that level of resilience and reliability. Be sure to consider those higher costs when making any comparisons.

What Is Five-Nines?

99.999% – A maximum downtime of approximately 5 minutes and 15 seconds per year.

Conclusion

The value gained by moving applications from the traditional enterprise platform onto the next “bright, shiny thing” rarely provides an improvement in the business process or the company’s bottom line. A great deal of analysis must be performed and each cost accounted for. Once the cost of the current platform is understood, the cost to move all applications must be identified. Each method of modernization has a defined cost model, and every application and database must be assigned to a methodology. This provides an idea of the cost to just move the application, then you must also obtain the true cost of running that application in the target environment.

Moving all the applications from their current platform has a complex algorithm. A typical equation for platform abandonment looks something like this:

- Cost for the modernization
- Cost of the time impact of the modernization
- Cost of the new platform (even if you have the base platform, you have to add capacity for these transformed applications)
- Cost of training existing traditional platform development/test/production staff
- Cost of new licenses and software (or additional costs of the new seats)
- Cost to train existing business users on the new interfaces
- Cost of disruption to services during transition
- Cost of failure of one or more of the transformed applications
- Cost to run a parallel operating environment during application transition

A more reasonable equation would be one for a multiplatform environment; placing the right application on the right platform. A typical equation for a multiplatform transformation looks like this:

- Cost to integrate existing development/test/production personnel
- Cost to integrate newer ITIL methodologies
- Cost to move appropriate parts of an application to another platform/technology

Every company has a different equation to understand the benefits of a platform abandonment program. Yet few companies can produce a ROI to justify completely abandoning a traditional platform.

Acronym Key and Glossary Terms

ADM	application development and maintenance
capex	capital expenditure
CDC	change data capture
CIO	chief information officer
I&O	infrastructure and operations
I/O	input/output
ISV	independent software vendor
ITIL	Information Technology Infrastructure Library
opex	operating expenditure
OSS	open-source software
ROI	return on investment
SLA	service-level agreement

Recommended by the Authors

[Advancing Technology in Multiplatform Modernization Tools](#)

[Application Modernization Should Be Business-Centric, Continuous and Multiplatform](#)

[Building a Multiplatform Application Modernization Business Case](#)

[Choose the Right Approach to Modernize Your Legacy Systems](#)

Recommended For You

[Bust Silos, Focus on Customers and Enhance Business Outcomes Through Value Streams](#)

[Application Modernization Should Be Business-Centric, Continuous and Multiplatform](#)

[How to Prioritize Application Inventory and Rationalization](#)

[Product-Outcome-Aligned Modernization Decisions \(Intrado\)](#)

[You've Built That Killer Application: Now What?](#)

© 2020 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner is a registered trademark of Gartner, Inc. and its affiliates. This publication may not be reproduced or distributed in any form without Gartner's prior written permission. It consists of the opinions of Gartner's research organization, which should not be construed as statements of fact. While the information contained in this publication has been obtained from sources believed to be reliable, Gartner disclaims all warranties as to the accuracy, completeness or adequacy of such information. Although Gartner research may address legal and financial issues, Gartner does not provide legal or investment advice and its research should not be construed or used as such. Your access and use of this publication are governed by [Gartner's Usage Policy](#). Gartner prides itself on its reputation for independence and objectivity. Its research is produced independently by its research organization without input or influence from any third party. For further information, see "[Guiding Principles on Independence and Objectivity](#)."