

# The Risks and Rewards of Modernization

NTT DATA





## Table of contents

Why modernize..... 03

### The six Rs

Resourcing ..... 04

Retiring ..... 06

Rehosting ..... 09

Recoding + Rewriting ..... 11

Re-architecting ..... 13

Conclusion ..... 16

Sources ..... 17

## Why modernize

Throughout government, aging workhorse systems and applications are in dire need of modernization. Recent events have served as a necessary catalyst to spur government into action.

To take full advantage of today's technological advancements that will transform systems and services for future government, modernization is an imperative.

Modernization has been a backburner issue for many government CIOs, who struggle to balance mounting technical debt with shrinking budgets while weighing the benefits of the evolving technology landscape. It's an endless balancing act as technical priorities teeter-totter with limited budgets and resources.

Increased demand for citizen services and other external economic and social pressures have created urgency for government agencies to modernize antiquated applications and infrastructure. New funding sources now provide the potential for government agencies to accelerate modernization.

The Technology Modernization Fund and the American Rescue Plan have allocated more than \$1 billion to fund modernization projects.<sup>1</sup> President Biden's May 2021 Executive Order on Improving the Nation's Cybersecurity has further amplified the call to modernize systems and applications throughout government.<sup>2</sup>

Despite the availability of funds, many government entities are still struggling to transform their organizations to take full advantage of cloud, security, technology and enterprise efficiencies, as well as savings of time, resources and investment.

In this ebook, we help you evaluate the risks and rewards of modernization. We'll examine modernization through the lens of the six Rs, which represent the primary decision paths and considerations for modernizing legacy systems.

1. Resourcing
2. Retiring
3. Rehosting
4. Recoding
5. Rewriting
6. Re-architecting

## Resourcing

The “silver tsunami” has hit the technical ranks of many in the government workforce. Individuals who are familiar with the code, patches, nuances and languages required to run aging systems are retiring from the labor force. Considering that many critical systems are 40-plus years old, it stands to reason that the expertise is aging, as well.

In April 2020, New Jersey Governor Phil Murphy issued an urgent plea, not for vaccines, but for COBOL programmers.<sup>3</sup> COBOL, the decades-old computer language that most university computer science programs no longer teach, was the stumbling block to processing unemployment benefits for the citizens of New Jersey.

So, the state put out a call for retired COBOL programmers to dust off their keyboards to help fix its 40-year-old unemployment system, which was identified as a major risk way back on the eve of Y2K. Sadly, New Jersey isn't alone. Many similarly aging systems are being maintained by an aging workforce. It's estimated that 220 billion lines of COBOL are still in use today.<sup>4</sup>

*StateScoop* reported in December 2020 that states are having to get more creative in developing

solutions to provide life support to aging systems. In Nebraska, state officials were forced to seek help from Latvia, a Baltic state in Northern Europe and one of the few remaining regions where mainframe technology is widely taught.<sup>5</sup> Not surprisingly, these new contractors are now supporting the state's unemployment insurance system.

**Working on ancient code has very little appeal to younger programmers, which magnifies the technical labor shortage.**

If you're trained in C# or Java, working on a COBOL mainframe is like stepping back in time. It's not all gloom and doom, however, as many large system integration companies have large teams that have been training in COBOL. As a result, they may better serve their clients. Organizations like COBOL COWBOYS<sup>6</sup> and the Open Mainframe Project<sup>7</sup> are seeking to fill the void by providing talent and developing open-source tools to better enable mainframe computing.

These measures alone can't support the 220 billion lines of old code scattered across the commercial and government sectors; the time is now to embrace more aggressive options to avoid further resource shortages clearly on the horizon.

## ! Risk

Critical government systems and the technical professionals who service them are both aging. Soon, the talent it takes to fix and maintain these systems may no longer be available. As a result, the risk of failure is great across key areas of security and reliability. Complete failure is also possible, as both systems and technologists age.

## ★ Reward

Moving to new technologies, infrastructure and applications opens the door for additional talent resources, capabilities and security. IT resources can focus on more pressing modernization tasks or implementing newer technologies.

Without the burden of legacy infrastructure and applications, IT leadership can concentrate their time and energy on improving citizen services, strategic priorities and mission execution.



## Retiring

Digital government is a ubiquitous term used throughout the information technology (IT) industry. It's everywhere, from online publications and congressional hearings to vendor websites and marketing brochures. Certainly, the collective goal of CIOs throughout state capitols and federal agencies is to enable digital government, so citizens can interact with government using the latest information and communication technologies.

We've become a digital nation. The voting public wants to interact with government from their mobile phones on social media and on easy-to-use websites quickly, efficiently and with a few clicks.

**Citizens want an experience like what they're accustomed to using in a commercial environment.**

Enabling and empowering digital citizens and enacting a future digital government requires modernizing legacy system architectures that simply can't accommodate the latest technologies. Modernization is often used as catch-all term, meaning different things to the government and different organizations.

True modernization is a comprehensive journey that considers a broader set of strategies. It's a complete vision of the end-state that includes migration, modernization and automation, while accelerating cloud-native solutions for an agile, cost-efficient enterprise. Many agencies are already modernized, having embraced the cloud first and then the cloud-smart policies of previous administrations, while others are still struggling with mainframe systems running on COBOL.

These groups must decide what to retire and what to keep running. Examining a complicated system pragmatically may be the best course of action. It's a matter of determining what represents the greatest risk and what will yield the greatest reward.

A host of other complications adds to the challenge and further confounds even the most dedicated CIOs. Many entrenched infrastructure systems are mission-critical and must remain functional while new replacement systems are designed and built. This lengthens countless modernization projects, many of which take years to complete. The Internal Revenue Service's Individual Master File (IMF) has been in service since the Eisenhower Administration.

Modernization efforts began on the 60-year-old system in 2009 and, according to the Government Accountability Office (GAO)<sup>8</sup>, replacing core functions may not be complete until 2030.<sup>9</sup> A modernization project that takes 21 years will be outdated before the modernization is complete.

The GAO's concern isn't unwarranted; the organization maintains a carefully curated list of critical federal systems in need of modernization. Disconcerting highlights from that list include Department of the Interior systems that support dams and power plants and a Department of Transportation system with valuable airport information.<sup>10</sup>

Although modernizing an infrastructure landscape is complicated, most agree it's time for government to retire the mainframe. To support agencies in this effort, Amazon Web Services (AWS) offers the AWS Migration Acceleration Program.

Through this program AWS, along with a select group of partners, helps government agencies reduce costs, boost productivity, improve operational resilience and increase business agility by moving workloads to the cloud. AWS suggests a straightforward three-step process that includes assessing readiness, mobilizing resources, as well as migrating and modernizing workloads. We'll walk through some of the important considerations as you evaluate modernization.

## Unemployment systems

Many Americans experienced first-hand the limitations of aging and slowing mainframes during the global pandemic.

The surge in unemployment claims caused many state mainframe systems to buckle under the demand.

Outdated systems couldn't keep up with demand, leaving as many as 10 million Americans waiting for their benefit checks.<sup>11</sup>

# The Risks and Rewards of Modernization

## Risk

Mainframes and outdated applications pose considerable risk to government operations and service delivery, including those related to reliability and security. In addition, these legacy technologies limit agility, scalability and responsiveness.

Maintaining old systems also represents significant hardware and software costs. Technical debt limits dollars for innovation and improving citizen services and doesn't allow government to take full advantage of the new technologies, applications and security measures vital to a digital economy.

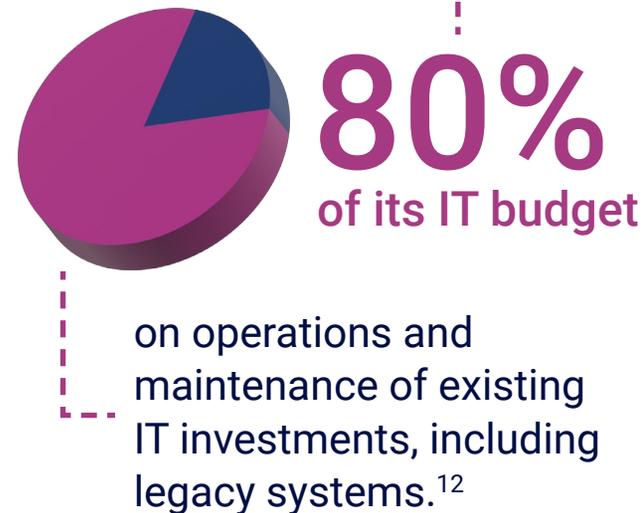
In our nation's capital, delays in modernization initiatives to fix the D.C. Department of Employment Services portal for submitting unemployment claims caused catastrophic delays in claims processing.

The system, built in early 2000s isn't mobile-compatible, making it inaccessible to many citizens.<sup>13</sup> If we've learned anything from the global pandemic, it's that systems must be built with flexibility to scale to accommodate the most unlikely of scenarios.

## Reward

Saying goodbye to the mainframe will allow government to realize savings of time, resources and dollars while positioning agencies to take full advantage of industrialized IT solutions available in the marketplace now and in the future.

The GAO reports that the federal government spends as much as



## Rehosting

One of the first steps in modernization may involve moving applications to the cloud. The quickest, easiest way to move to the cloud is through rehosting (also called re-platforming or lift and shift) your existing applications in the cloud.

Existing applications and data are redeployed in a like-for-like fashion, retaining the existing source code and functionality. Lift and shift typically migrates on-premises applications to the cloud with little or no changes to the applications. Many consider rehosting the easy way to move to the cloud. Rehosting can happen quickly and with minimal disruption and planning.

A lift and shift approach solves many concerns of scalability and is often the best first step in modernization, but it by no means represents a fully modernized cloud-native solution created to take full advantage of the features and functions required to meet the demands and challenges of government CIOs. Most public sector CIOs believe cloud adoption acceleration is business critical. According to a recent National Association of State Chief Information Officers (NASCIO) study,

**83% of respondents feel it's essential to fuel innovation and new business models.<sup>14</sup>**

When considering rehosting, it's important to select a vendor with experience and competency in moving applications to the cloud. For example, the business rigor required to achieve and maintain AWS Mainframe Competency provides a solid industry accreditation, so government organizations can confidentially select a vendor with rehosting experience and a proven track record of success.<sup>15</sup>

Rehosting assumes that the applications satisfy the needs of the business. Often, savings from rehosting can offset the cost of future modernization. Rehosting applications may also involve a small amount of up-versioning. In this case, organizations should consider an application architecture diagram or an on-premises application.

When rehosting, it's best to have evaluated the architecture of the application and determined a future phase plan, for each application tier, on what can be replaced with a cloud-native service.

## Risk

With a comprehensive plan and an application roadmap, the risk associated with rehosting in the cloud is minimized. Although rehosting doesn't take advantage of cloud-native benefits such as elasticity and cost optimization features, it's a step in the right direction.

The Federal Information Technology Acquisition Reform Act (FITARA) provides a scorecard for agencies across government that ranks several critical IT categories, one of which is Modernizing Government ACT. Nine of the top 25 agencies received a grade of C, with two barely passing with D grades.<sup>16</sup> Critical government systems with failing IT grades certainly represent risk.

## Reward

With rehosting, organizations can unlock immediate savings in hardware costs and software licensing fees. Organizations that rehost often see higher uptime, lower maintenance, better performance and labor savings. Agencies can save 30% to 70% by merely rehosting. While workloads are migrated to the cloud, agencies should adopt DevSecOps

practices and processes like continuous integration and continuous delivery (CI/CD) and SAFe. This can be an effective way to launch a two-part strategy that begins with lift and shift to economize and realize savings, and then uses those savings to incrementally fund re-engineering initiatives.

## Mainframe rehosting with UniKix

-  **Lowers annual IT operating costs**
-  **Reduces migration risk, time and complexity**
-  **Simplifies IT environments to accommodate growing workloads and new demands**
-  **Provides a native, mainframe compatible processing environment on open systems**
-  **Delivers a proven platform for evolving technologies such as cloud computing, mobility and virtualization**

## Recoding + Rewriting

When evaluating an IT environment from a modernization perspective, rewriting and recoding go hand in hand. Both benefit the system once applications are operational in the cloud.

Many organizations choose the all-in approach to modernization by recoding or rewriting applications at the same time as they move to the cloud.

Recoding converts legacy code to a more modern language, such as moving from COBOL to Java or C#. Typically, functionality is retained, but the resulting code is easier to maintain and enhance. Automated software tooling is available to help, converting one programming language to another (retaining functionality).

Rewriting an application means starting from scratch using traditional methods. Many organizations attempt to rewrite legacy systems through traditional Waterfall and Agile methodologies. They effectively gather requirements as if gathering them for a greenfield development project prior to attempting to rewrite the systems.



## Risk

The challenge is maintaining and enhancing the resulting system. Code conversion generally replaces the existing source code on a line-by-line basis. This approach implies that a COBOL program converted to Java will retain the procedural aspects of the COBOL program but be written in Java syntax. The resulting solution will require skills in both COBOL (to understand the program process flow) and Java (to understand the syntax).

Syntax changes sometimes fail to take full advantage of the benefits of the new language. In addition, many organizations that take this approach spend several years remediating converted programs into a more elegant or perfect form of Java or C#.

In some cases, remediating involves making assumptions and/or guesses about the functionality of existing, poorly documented code. And it frequently results in piling complexity on top of an already-complex application set. This code bloat can inhibit future application enhancements.

Projects that involve rewriting applications often fail and/or experience huge delays, subsequently incurring major cost overruns because the gathered requirements don't represent all the functionality

embodied in the original system. Many manual conversions fail to effectively understand the complex relationships to business logic and the associated data elements typically discovered late in the project execution, making it a costly mistake.

## Reward

The benefits of code conversion are like rehosting: The mainframe is left behind, expensive software license fees are eliminated and a more modern infrastructure is implemented. And if modern tools are implemented, the solution can be easier to enhance and maintain in the future. Rewriting can produce crisp code that takes advantage of features in the language and the cloud. The new application includes all the desired features and functionalities.

## Re-architecting

Re-architecture extracts legacy knowledge by matching or enhancing functionality by using advanced tooling and applying best practices in developing modern architecture.

Sometimes referred to as refactoring, re-architecture retains the best features of the original application while removing unnecessary functionality and taking advantage of new infrastructure. Inherent to this approach is modernization to a more robust programming language like Java or C#.

**Re-architecture takes full advantage of cloud-native benefits such as security, scalability and speed.**

It's also possible to include a new, modern front-end for application services using low-code/no-code tools. This option affords agencies the benefit of a user-friendly interface that helps ensure end-user satisfaction while reducing the application's resource drain on the agency.

While every agency will have a mix of options and workloads in its portfolio, it's advisable to conduct a thorough analysis before you begin, assessing the right path for each legacy application.

### Risk

Refactoring takes time and requires a comprehensive future-state architecture and a strategic plan to achieve that architecture. In many cases, this can be the longest path to exiting the mainframe.

Refactoring extends exposure to two of the greatest risks to success – fitting within budget constraints because mainframe costs aren't reduced until the late stages of the project and potentially extending exposure to security vulnerabilities.

Legacy systems represent a significant security risk. Older systems and applications simply don't have the security protections and protocols necessary to prevent today's sophisticated attacks.

Nefarious actors target weaknesses, creaky old architectures and applications that wear a bullseye and have the potential to expose critical infrastructure supported by the government. This exposes the personally identifiable information of millions of citizens. Re-architecture helps create a system while building a solid defense.

## ★ Reward

With re-architecture, government maximizes the inherent cost savings of cloud, as well as provides a platform for digital innovation and operational excellence. While this approach demands a greater up-front investment, agencies benefit from the security features of immutable infrastructure, auto-scaling, greater elasticity and more.

Modern systems feature modern security, enabling agencies to meet the expectations of the Biden Administration's Executive Order by adopting security best practices, Zero Trust architectures, secure cloud services, and analytics for identifying and managing cybersecurity risks.

While many mainframe systems feature outdated technologies with known vulnerabilities, cloud systems are updated continuously to protect against the latest threats.

With the cloud's shared responsibility model, agencies are assured that cloud providers secure base cloud functionalities. AWS, for example, maintains responsibility for security from its host operating system (OS) and virtualization layer.

Users are responsible for everything from the guest OS up to and including the firewall. Cloud providers also help ensure security best practice and guideline compliance, including NIST's Cloud Security Framework (CSF) and FedRAMP.



## The United States Air Force's modernization success

The United States Air Force (USAF) needed to modernize its 50-year-old legacy mainframe supply chain platform to reduce operating costs, improve system agility and ensure full mission readiness of its bases around the world.

Through a partnership spanning nearly 20 years, the USAF and NTT DATA modernized software development methods, re-platformed a complex proprietary system and pivoted a mission-critical enterprise logistics platform to the USAF cloud.

### Updated system features

- Supported \$18 billion in inventory for over 100,000 direct and indirect users worldwide
- Saved \$25 million annually on hosting costs
- Accelerated platform modernization by transitioning from Waterfall to Agile development

- Transformed the system from a customized COBOL on Unisys platform to a Java on Red Hat Enterprise Linux environment
- Migrated a hierarchical database to an Oracle relational database
- Eliminated the USAF's footprint on a cluster of five legacy mainframes
- Migrated a 120-plus server, 5 TB production environment to AWS GovCloud/USAF Cloud
- Expanded to monthly software releases using an Agile delivery model
- Ensured future agility with security through DevSecOps

## Conclusion

To build the government of the future, we can't rely on applications and infrastructures of the past. The rapid advancement of technologies, coupled with increasing citizen demand for digital services, means modernization efforts must be accelerated.

In a recent article titled "Modernizing the Monolith," Dereck Magill, executive director of the Association of High Computing Professionals, provided sound advice to CIOs struggling with legacy systems:

**"Gather internal knowledge, obtain the appropriate technology and process skills, and determine the best path for your specific monolithic application based on cost-benefit analysis, and you will be ready to start the journey to monolith modernization."**<sup>17</sup>

The task ahead seem daunting, but careful strategy and planning weighed against the risks and rewards will pave the road for the government of the future.



Visit our [website](#) to learn more about the risks and rewards of modernization.

# The Risks and Rewards of Modernization

## Sources

- 1 The Technology Modernization Fund's American Rescue Plan. <https://tmf.cio.gov/arp/>
- 2 The White House. "Executive Order on Improving the Nation's Cybersecurity." May 12, 2021 <https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/>
- 3 New Jersey Governor Phil Murphy press conference. April 5, 2020. <https://www.youtube.com/watch?v=HSVgHISTPYQ&t=3s>
- 4 M. Lynn. "Open Mainframe Project Helps Fill the Need for COBOL Resources." Open Mainframe Project. April 9, 2020. <https://www.openmainframeproject.org/blog/2020/04/09/open-mainframe-project-helps-fill-the-need-for-cobol-resources>
- 5 Colin Wood. "With mainframe support scare, North Dakota turns to Latvia." StateScoop. December 28, 2020. <https://statescoop.com/north-dakota-mainframe-latvia-workforce/>
- 6 COBOL COWBOYS. Professionals for Legacy COBOL Systems. <http://cobolcowboys.com/>
- 7 Open Mainframe Project. <https://www.openmainframeproject.org/>
- 8 Government Accountability Office. "Information Technology: Cost and Schedule Performance of Selected IRS Investments." Report to Congressional Committees. October 2021. <https://www.gao.gov/assets/gao-22-104387.pdf>
- 9 Natalie Alms. "IRS tax system modernization falls further behind." Government Computing News. October 26, 2021. <https://gcn.com/articles/2021/10/26/irs-modernization-delays.aspx>
- 10 Republican Policy Committee. "Modernizing Government Information Technology." May 15, 2020. <https://www.rpc.senate.gov/policy-papers/modernizing-government-information-technology->
- 11 Amy Fontinelle. "Why Your Unemployment Check or Direct Deposit Is Late (and What to Do About It)." Investopedia. September 11, 2021. <https://www.investopedia.com/why-your-unemployment-check-or-direct-deposit-is-late-and-what-you-should-do-5186560>
- 12 Government Accountability Office. "Information Technology: Agencies Need to Develop and Implement Modernization Plans for Critical Systems." Testimony before the Subcommittee on Emerging Threats and Spending Oversight Committee on Homeland Security and Governmental Affairs, U.S. Senate — Statement of Kevin Walsh, Director, Information Technology and Cybersecurity. April 21, 2021. <https://www.gao.gov/assets/gao-21-524t.pdf>
- 13 Fenit Nirappil and Darran Simon. "Delays in overhauling D.C. employment site add to turmoil of coronavirus layoff." Washington Post. April 2, 2020. [https://www.washingtonpost.com/local/dc-politics/delays-in-overhauling-dc-unemployment-site-fueled-turmoil-in-covid-19-crisis/2020/04/02/ebe7d81e-73ad-11ea-87da-77a8136c1a6d\\_story.html](https://www.washingtonpost.com/local/dc-politics/delays-in-overhauling-dc-unemployment-site-fueled-turmoil-in-covid-19-crisis/2020/04/02/ebe7d81e-73ad-11ea-87da-77a8136c1a6d_story.html)
- 14 Accenture and NASCIO. "A Fresh Look: Capitals in the Clouds. 2021. <https://www.nascio.org/wp-content/uploads/2021/10/NASCIO-Accenture-Cloud-Study.pdf>
- 15 Renata Melnyk, Ramesh Jadhav and Phil de Valence. "Introducing the New AWS Mainframe Migration Competency Featuring Validated AWS Partners." AWS News Feed. March 31, 2021. <https://awsfeed.com/whats-new/apn/introducing-the-new-aws-mainframe-migration-competency-featuring-validated-aws-partners>
- 16 FITARA. "Committee on Oversight and Reform Biannual Scorecard." MeriTalk FITARA Dashboard. July 2021. <https://fitara.meritalk.com/assets/pdf/scorecard-12-methodology.pdf>
- 17 Derek Magill. "Map Your Journey to Monolith Modernization." The New Stack. April 14, 2021. <https://thenewstack.io/map-your-journey-to-monolith-modernization/>