From COBOL to Cloud

Tackling Mainframe to Cloud Migration in the Federal Government

Authors

Doug Manning
GovCIO

Kory Caze
TSRI

Transforming Government IT
govcio.com

It may seem odd to still be discussing mainframes in 2022. However, while Cloud has taken center stage in Federal IT, legacy mainframes continue to be integral to many mission-critical government systems, data centers, and environments.

Despite their age, mainframes are still good at doing their job- hosting numerous critical enterprise applications and handling large volumes of concurrent users. Customers who depend on mainframes also fear losing the decades worth of valuable data mainframes often store if a modernization effort fails.

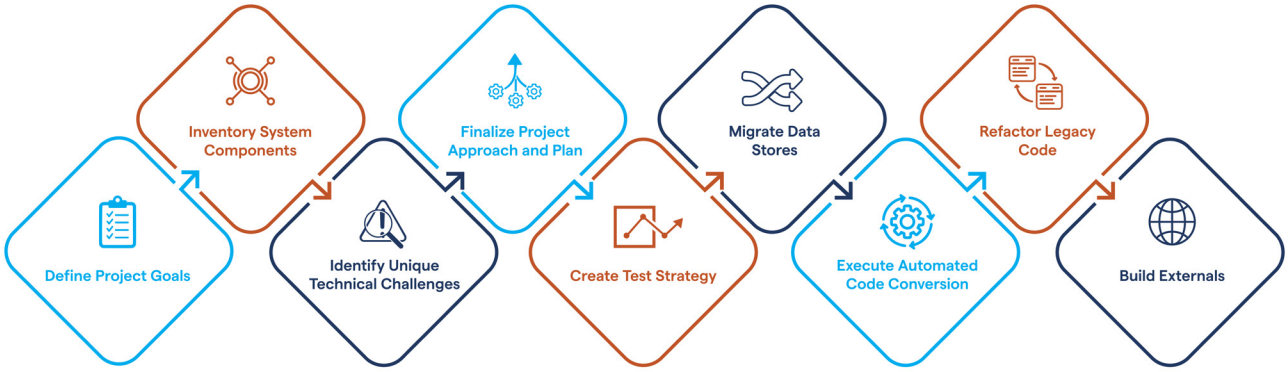
Nevertheless, the value mainframes bring cannot outweigh the issues. Even with updates, the programming languages on which mainframes run, such as COBOL, are outdated; they have limited legacy functionality and lack the flexibility, scalability, and modern features, such as microservices and containerization, of more recent platforms. Mainframes also incur significant maintenance costs and, most significantly, the workforce with the skillsets to sustain mainframes is dwindling, meaning that, sooner or later, federal agencies must transition away from them.

GovCIO has worked extensively to establish a holistic mainframe migration process for our federal customers that improves organizational performance while minimizing customer risk.

Read on to learn how GovCIO works with customers to rapidly migrate them from mainframes to modern cloud architectures that can evolve with them, all while preventing negative impacts to their mission, day-to-day operations, and security.

The Mainframe to Cloud Migration Process

Team GovCIO's Mainframe to Cloud Migration Process



1. **Define Project Goals:** GovCIO teams first work with all stakeholder groups to reach a consensus on project goals and validate goals are realistic. Too often projects fail because the scope is too broad. Although it's tempting to want to modernize legacy business processes in parallel with technology, the availability of stakeholders and decisionmakers often creates significant delays and ultimately increases the risk of project failure.

2. **Inventory System Components:** Based on the project goals and scope, GovCIO creates a comprehensive system component inventory, including source code, job streams, interfaces, and externals. We examine available resources to determine system usage, performance metrics, and possible obsolescence. Where possible, our teams analyze system logs to verify what software is still used and whether outputs are still relevant. Oftentimes, 30-40% of production software is no longer relevant and can be retired. Using automated tools and log analysis, we minimize the time spent performing this analysis but

ensure we can remove any obsolete components from the project scope.

3. **Identify Unique Technical Challenges:** Collaborating with stakeholders, we identify each unique technical challenge the project presents and examine possible solutions for the target platform. Challenges may include assembler language subprograms, unique operating system calls, legacy APIs, and machine dependent utilities. We evaluate all data storage and database environments to build initial strategies for the target platform. Finally, our team examines user interface requirements and existing interface technologies to determine what can be used going forward and what must be replaced.

4. **Finalize Project Approach and Plan:** Given the overall goals, cost, schedule, and risks, the project team can now agree on the optimal way forward. Again, it's important to keep project goals realistic and not attempt to solve more problems than necessary. GovCIO applies an iterative, Agile execution approach for migration projects. This gives us the flexibility to adapt to change; the ability to deliver functionality in small increments, which improves customer alignment and reduces risk; and allows for continual improvement, increasing our likelihood of project success.

5. **Create Test Strategy:** To reduce the risk of a new system leaving out any needed legacy functionality, GovCIO's testing strategy focuses on verifying the target implementation performs the same functions as the legacy system, making sure to consider all interfacing systems. Prioritizing this makes our testing strategy much less complex than if we were to verify the new system does everything it used to do, as well as the functional changes that have been introduced in parallel with the technology migration. Our team advocates the use of automated testing tools so we can perform functional and performance testing as often as possible, ideally with each system build.

6. **Migrate Data Stores:** Using an automated, repeatable, and verifiable process, GovCIO migrates legacy platform data to the target implementation in parallel with the code conversion. This ensures we can effectively migrate straightforward data stores such as relational databases and more antiquated, and complex, hierarchal or network databases. With non-database file structures, such as indexed sequential files, we transform them into structures more appropriate for the target architecture.

7. **Execute Automated Code Conversion:** To convert code quickly and efficiently, Team GovCIO strives to apply a 99.9% automated approach that converts the entire codebase each time a transformation is performed. Team GovCIO partnered with The Software Revolution, Inc. (TSRI), which provides automated application modernization and refactoring, to lead this phase of such projects. Automated modernizations are critical for systems with frequent software updates to support on-going functional requirements. Conversion problems identified during testing are not corrected in the converted code but applied to the conversion engine configuration until all desired results are obtained. This greatly reduces the time needed to "freeze code" while the system is migrated.

8. **Refactor Legacy Code:** As we move through the conversion process, it is likely we'll find faulty legacy code, such as latent defects, platform specific dialects of SQL, arcane data types and file formats, none of which will convert cleanly to the target platform. If this happens, we work with TSRI to implement its automated refactoring engine to update affected code in the legacy code base and then re-run through the conversion process. When feasible we release legacy code to the production system. This simplifies configuration management and keeps the codebase undergoing conversion synchronized with the production baseline.

9. **Build Externals:** External components, identified during the system inventory, are modules for which no legacy code previously existed. Our conversion process simply "stubs out" the calls to these modules within converted code. The project team determines if the module function is still relevant on the destination platform and, if so, the best way to replace the functionality in the converted system. It's possible these modules can be replaced using libraries on the destination platform, or open-source APIs. Modules without available replacements are built using the appropriate coding language.

Team GovCIO's Success Story at

Housing and Urban Development (HUD)

Overview

Over the past few decades, a growing challenge within the government space has been the continued use of mainframes that ran decades-old applications, some of which had been written in customized languages that fit the agency's mission. One such federal agency that faced this challenge was The Department of Housing and Urban Development (HUD).

The Technology Modernization Fund (TMF), authorized in 2017 and disbursed by the General Services Administration (GSA), provided HUD with the opportunity and resources to modernize many of its critical applications that ran on a mainframe.

The Challenge

To get allocated funds from GSA's TMF, HUD needed a proven contracting team to deliver a feasible solution for modernizing its critical legacy applications. These applications, which ran on the same Unisys mainframe and were written in COBOL code, included:

- **The Computerized Home Underwriting Management System (CHUMS)** - Supports Home Ownership Centers (HOCs) staff in the processing of single-family mortgage insurance applications, from initial receipt through endorsement. This system conducts thousands of transactions with external resources daily.
- **Credit Alert Verification Reporting System (CAIVRS)** - Used by multiple federal agencies to determine if a loan applicant has outstanding federal debt that is in default or foreclosure status.
- **The Line of Credit Control System (LOCCS)** - A disbursement and cash management system that distributes over $28 billion annually for grants and subsidies.

GovCIO's Solution

Team GovCIO worked extensively to establish a holistic mainframe migration process for our federal customers that improves organizational performance while minimizing customer risk. After identifying HUD's need, GovCIO (formerly Salient CRGT) partnered with TSRI to proactively develop a working prototype for HUD that demonstrated the efficacy and viability of our process. HUD was able use this prototype to support its application for TMF funding. Once granted the funding, HUD awarded a sole-source contract to Team GovCIO. For this effort, HUD had five key goals:

- Shutdown the Unisys Mainframe.
- Complete in two years or less.
- Prevent disruption to critical business functions.
- Support ongoing updates and enhancements, including legislative and regulatory changes in parallel.
- Minimize the need for program area decisions.

Using our nine-step process, described in detail earlier in this post, Team GovCIO successfully worked with HUD to rapidly migrate them from their COBOL-based mainframes to a modern Azure-based cloud service that can evolve with them, all while preventing negative impacts to their mission, day-to-day operations, and security.

9 Steps from COBOL to Cloud

1. Define Project Goals
2. Inventory System Components
3. Identify Unique Technical Challenges
4. Finalize Project Approach and Plan
5. Create Test Strategy
6. Migrate Data Stores
7. Execute Automated Code Conversion
8. Refactor Legacy Code
9. Build Externals

Results

The project was completed in two phases. Phase one comprised the Housing systems, CHUMS and CAIVRS. Phase-two covered the CFO system, LOCCS.

While our team planned the phase-one go-live for a three-day holiday weekend to allow the most time for migration and any unforeseen challenges, we completed the system and data migration in just 14 hours. The go-live decision was made mid-day on the second day and the system went live on schedule with a normal production workload. The phase two migration was planned for a normal two-day weekend and was successfully completed in under 24 hours.

On day one of the system going live, CHUMS supported 25,356 users and 299,715 transactions with only three user problems reported. In the first 30 days after the LOCCS migration from the Unisys to Azure, the system disbursed just over 2.7 billion dollars in program funds without error.

The successful completion of the automation modernization project solved one of the longest standing and most complex IT challenges facing HUD – moving these mission critical systems off the legacy mainframe platform.

As HUD was among the first to make use of the TMF funding for this type of initiative, HUD's success can also be viewed as a significant endorsement of what agencies can achieve with TMF funding. Team GovCIO proved a rapid, highly accurate migration of mainframe systems to Cloud services is possible, even for critical complex projects such as HUD's, using the TMF.

Final Thoughts

Overcoming the risks and complexities of migrating mainframes to Cloud-based environments can be incredibly daunting in the Federal space. By fully grounding our approach in sound engineering principles and repeatable, verifiable automated processes, GovCIO and our partner TSRI have built a mainframe migration process that minimizes risk and defect rates and maximizes successful outcomes for our clients.

Doug Manning, Operational Vice President, GovCIO
With 38 years supporting the Federal Market Doug has supported modernization efforts on many platforms and customers. The HUD TMF project was especially gratifying in that HUD was a longtime client and Doug led the effort from inception to completion.

Kory Caze, Lead Account Executive, TSRI
After a career in the U.S. Army, Kory has led account relations for a majority of TSRI's federal government and DoD projects for the past five years. Kory has been an integral part of the HUD TMF projects for the past four years.