



---

## The Roadmap to Legacy System Modernization: Phased Approach to Mainframe Migration and Cloud Adoption

Srinivas Adilapuram

Senior Application Developer, ADP Inc, USA

---

**Abstract:** Legacy systems often present significant challenges for organizations seeking to modernize their IT infrastructure. Migrating these systems to contemporary platforms requires careful planning and execution to mitigate risks and maximize benefits. This paper addressed the challenges of migrating legacy mainframe systems to modern cloud platforms. It proposed a phased approach to mitigate risks such as performance issues, skill gaps, and high costs. The study outlined three phases: assessment and planning, rehosting and refactoring, and application modernization. Recommendations included employing CI/CD pipelines, upskilling teams, and utilizing performance monitoring tools.

**Keywords:** Legacy modernization, mainframe migration, cloud computing, Spring Boot, security, CI/CD

---

### 1. Introduction

Modernizing legacy mainframe systems is a critical undertaking for organizations seeking to enhance agility, reduce costs, and leverage the benefits of cloud computing. These systems, often decades old, present unique challenges due to their monolithic architecture, specialized skillsets required to maintain them, and potential risks associated with migration. A poorly planned transition can disrupt critical business operations, introduce performance bottlenecks, and expose vulnerabilities.

To overcome these challenges, organizations must adopt a phased approach to modernization [1]. This involves assessing the current system landscape, identifying critical workloads, and determining the best modernization strategy—whether rehosting, re-platforming, or refactoring. Advanced tools and methodologies, such as automated code analysis and containerization, can streamline the process while mitigating risks [2]. Moreover, integrating robust security measures during migration ensures that sensitive data remains protected throughout the transition.

This paper explores a phased approach to mainframe migration, mitigating these risks through careful assessment, strategic rehosting, and incremental modernization. It examines key considerations for each stage, emphasizing best practices for ensuring a smooth and successful transition to a modern, cloud-based environment. This includes detailed recommendations for utilizing modern tools and technologies, such as AWS Mainframe Modernization for rehosting and Google Cloud's Secret Manager for securing sensitive data within modernized applications.

Furthermore, the paper addresses the critical aspect of upskilling existing personnel. Equipping teams with the necessary knowledge and expertise in cloud technologies and modern programming languages is essential for successful modernization and ongoing maintenance. Organizations can empower their workforce to effectively manage and optimize their modernized applications by investing in training and development that will ensure long-term success in cloud applications.

### 2. Literature Review

Modernizing legacy systems is a critical undertaking for organizations seeking to enhance agility, reduce costs, and improve operational efficiency. Mainframe migration is a key aspect of this process that involves moving



applications and data from outdated mainframe environments to modern platforms. A phased approach to mainframe migration offers a structured and manageable way to achieve this transition, minimizing disruption and mitigating risks. This literature review analyzes research related to the topic, exploring key challenges, strategies, and best practices.

Migrating legacy systems presents numerous challenges. Johnson et al. (2018) highlighted the complexities associated with migrating legacy systems to the cloud, including data compatibility issues, security concerns, and the need for extensive testing [2]. The study underscores the need for careful planning and execution to ensure successful migration initiatives.

A phased approach to mainframe migration offers several advantages. Smith (2019) advocated for phased migration strategies, allowing organizations to incrementally transition applications and data, reducing the impact on business operations [1]. This approach enables better risk management, as potential issues can be identified and addressed in earlier phases. Moreover, phased migrations allow for continuous testing and validation, ensuring the integrity and functionality of migrated applications.

Various techniques and best practices facilitate mainframe modernization. Lee (2019) presented a case study on automating legacy application refactoring, demonstrating how automation can accelerate the modernization process and improve code quality [5].

Containerization has emerged as a valuable tool in mainframe modernization. Davis and Chen (2019) discussed the role of containerization in packaging and deploying applications, enabling portability and scalability [3]. This technology facilitates the transition to modern infrastructure, including cloud environments. Furthermore, Patel and Singh (2017) explored the evolution of hybrid architectures in legacy system modernization, where organizations leverage a combination of on-premises and cloud-based resources to optimize performance and cost efficiency [4].

Cost optimization is a crucial aspect of mainframe migration projects. Organizations need to carefully assess the costs associated with migration, including hardware, software, and personnel, to ensure a positive return on investment. Taylor (2018) emphasized the importance of security considerations in legacy system modernization, addressing potential vulnerabilities, and implementing appropriate security measures to protect sensitive data [6].

Agile methodologies can enhance the effectiveness of mainframe migration projects. Zhao (2019) discussed the application of agile principles in mainframe migration, promoting iterative development, continuous feedback, and collaboration among stakeholders [7]. Agile approaches enable flexibility and adaptability, allowing organizations to respond to changing requirements and challenges throughout the migration process.

Modernizing legacy systems through a phased approach to mainframe migration is essential for organizations seeking to remain competitive in today's digital landscape. This literature review has highlighted key challenges, strategies, and best practices associated with this process. Organizations can successfully transition to modern platforms, unlocking the benefits of enhanced agility, reduced costs, and improved operational efficiency.

### **3. Problem Statement: Traditional Mainframe Migration**

Mainframes have historically provided robust and reliable computing power for critical business operations; organizations increasingly recognize the need to modernize their legacy systems.

Migrating mainframe workloads to modern platforms, such as cloud-based or hybrid environments, offers numerous potential benefits, including reduced costs, increased agility, and access to a wider range of technologies. However, this modernization process is not without its challenges. This section examines the key problems associated with mainframe migration, focusing on the technical and organizational hurdles that organizations must overcome to ensure a successful transition.

#### **Technical Challenges in Mainframe Migration**

Migrating mainframe workloads to modern platforms involves addressing several intricate technical challenges that can impact the success of the project. One of the foremost difficulties arises from the fact that mainframe applications are often built using legacy programming languages, such as COBOL, which are not easily adaptable to contemporary cloud-based or hybrid infrastructures. These legacy languages typically lack the extensive libraries, frameworks, and support available for modern programming languages, requiring either a complete rewrite of applications or complex conversion processes to make them functional on new platforms.





Figure 1: Data migration concept image

Data migration also presents a significant hurdle. Mainframe databases, like IBM DB2, use proprietary data formats that may not be directly compatible with newer database systems like SQL or NoSQL databases. As a result, organizations must carefully plan how to migrate the data, ensuring that data integrity is maintained throughout the process. This may involve mapping old data structures to new ones, which can be both time-consuming and error-prone, especially for large-scale systems with millions of records.

Additionally, performance optimization in modern environments can be challenging. Mainframe systems were designed to handle massive transaction volumes and perform complex computations in a highly efficient manner. When migrating to a cloud-based or hybrid environment, organizations often struggle to replicate the high level of performance that mainframes provide.

Differences in hardware architecture, the virtualized nature of cloud platforms, and even the data access patterns can lead to performance degradation. This can especially affect applications that rely on real-time processing or high throughput, requiring a careful tuning of resources in the new environment. To mitigate these technical issues, organizations must work with experts who understand both the intricacies of the mainframe and the capabilities of the new environment to ensure a seamless transition.

#### **Organizational Challenges in Mainframe Migration**

While the technical complexities of mainframe migration are considerable, organizations also face significant organizational hurdles when modernizing legacy systems. One of the most pressing challenges is the shortage of skilled personnel who are proficient in both legacy mainframe technologies and modern computing environments. Mainframe systems, which have been in place for decades, often require specialized knowledge that few current IT professionals possess, making it difficult for organizations to find talent capable of managing the transition effectively. This talent gap can lead to prolonged migration timelines, increased costs for hiring and training, and potential delays due to a lack of expertise on the project team.

Another critical organizational challenge is resistance to change, which is often rooted in the fear of disruption. Employees who are accustomed to the stability and familiarity of mainframe systems may be hesitant to embrace new technologies and processes. Furthermore, there may be concerns over job security, as the migration to modern platforms can sometimes lead to a reduction in the need for roles that support legacy systems. This resistance can manifest in a lack of buy-in from key stakeholders, particularly from the IT department and other critical teams, which can create roadblocks in the migration process.

#### **Inherent Risks in Mainframe Migration**

Mainframe migration is not without its risks, and organizations must carefully assess and mitigate these risks throughout the process to ensure a successful outcome. One of the most significant risks is the potential for performance bottlenecks if the target environment is not properly configured or optimized. Mainframe systems



were built to handle specific workloads, often tailored to a company's unique business processes. When migrating to a new platform, performance optimization becomes critical, especially for mission-critical applications. Failing to properly tune the cloud or hybrid environment can lead to slower performance, downtime, or even service disruptions, which can negatively affect business operations.

Another risk involves the skill gaps within the organization. If there is insufficient expertise in both the legacy and modern technologies, the migration process can be delayed or flawed. Inadequate planning or mismanagement during the migration can lead to costly errors, potentially disrupting business continuity. This can also result in significant financial losses, as companies may have to rework parts of the migration or resolve unanticipated issues, further escalating the project's cost.

Cost overruns represent another key risk in mainframe migration projects. While the long-term benefits of migrating to modern platforms can be substantial, the upfront costs can be significant. These expenses may include hardware, software, cloud services, consulting fees, and training for employees to manage the new systems. If the migration is not properly scoped and managed, these costs can escalate quickly, impacting the organization's financial stability. Furthermore, if the migration project does not meet expectations, it could lead to project failure, which could result in business disruptions and irreparable reputational damage.

#### 4. Solution: Phased Approach To Mainframe Migration

Modernizing legacy mainframe systems is a complex undertaking, often fraught with challenges such as tightly coupled architectures, outdated programming languages, and a lack of skilled personnel. A phased approach allows organizations to mitigate these risks, minimize disruption, and incrementally transition to a modern, agile environment.

This section outlines a three-phased strategy for mainframe migration, leveraging cloud technologies and modern software development practices.

##### Phase 1: Assess and Plan

The first phase focuses on thorough assessment and planning. This involves a comprehensive evaluation of existing mainframe workloads, identifying critical systems and dependencies, and understanding the business processes they support. Data migration strategies are defined, considering data volume, sensitivity, and regulatory requirements. A detailed roadmap is created, outlining migration timelines, resource allocation, and potential risks.

This phase also includes a technology assessment. Organizations evaluate cloud platforms like AWS, Azure, or GCP, considering factors like cost, scalability, and security. Modernization tools, such as AWS Mainframe Modernization service, are explored to facilitate automated code conversion and rehosting.

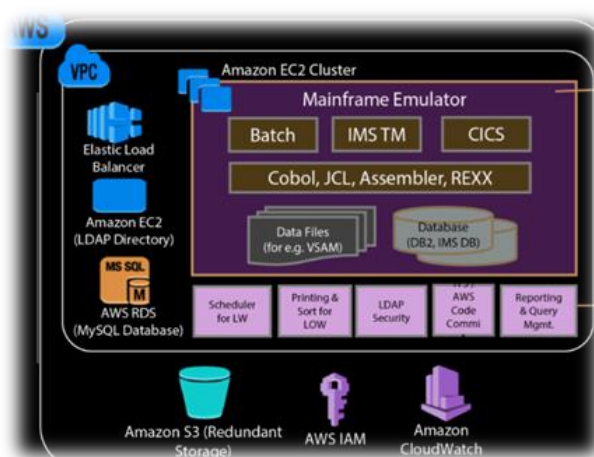


Figure 2: Amazon Mainframe Modernization Service

##### Phase 2: Rehost and Refactor

The second phase involves rehosting and refactoring. Rehosting, often referred to as "lift and shift," involves migrating applications to the cloud with minimal code changes. This approach provides a quick win, enabling



organizations to leverage cloud benefits like scalability and cost-efficiency without significant code modifications. Tools like AWS Mainframe Modernization can automate this process, converting legacy code to run on cloud-native platforms.

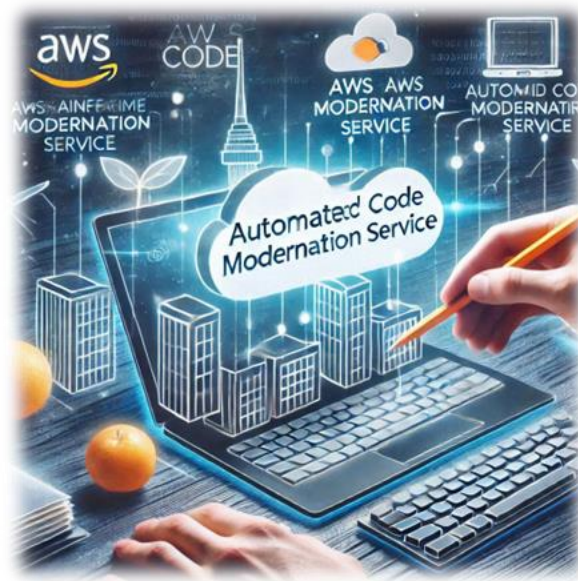


Figure 3: AWS Automated Code Moderation

Refactoring, on the other hand, involves restructuring and optimizing code to improve maintainability and performance. This might include breaking down monolithic applications into microservices, adopting modern programming languages like Java or Python, and leveraging cloud-native services like databases and message queues.

Consider a scenario where a mainframe application manages customer accounts. During rehosting, the application might be migrated to a cloud virtual machine with minimal changes. In the refactoring stage, this application could be decomposed into microservices, with separate services for account creation, updates, and transactions.

Tools like AWS Mainframe Modernization offer command-line interfaces to initiate and manage migrations for rehosting scenarios.

### Phase 3: Modernize Applications

The final phase focuses on modernizing applications to fully leverage cloud capabilities and agile development practices. This involves re-architecting applications using modern frameworks like Spring Boot for backend development and Angular for frontend development. This enables organizations to build responsive, scalable, and maintainable applications that meet evolving business needs.

This phase also includes adopting DevOps practices and implementing Continuous Integration/Continuous Delivery (CI/CD) pipelines to automate build, testing, and deployment processes. This ensures rapid iteration and frequent releases, enabling organizations to deliver value faster and respond to market changes quickly.

For example, the customer account management application, now refactored into microservices, could be further modernized with a Spring Boot backend and an Angular frontend.

CI/CD pipelines would automate the deployment of microservices, enabling rapid updates and feature releases.

### Recommendations for Successful Migration

To ensure a smooth and successful mainframe migration, consider the following recommendations:

**Incremental Migration:** Adopt a phased approach, migrating applications incrementally to minimize disruption and manage risks effectively.

**CI/CD Pipelines:** Implement CI/CD pipelines to automate the build, testing, and deployment of modernized applications, enabling rapid iteration and frequent releases.



**Upskilling:** Invest in upskilling existing teams on cloud technologies, modern programming languages, and DevOps practices. This ensures that the organization has the necessary skills to support the modernized environment.

**Monitoring and Optimization:** Utilize monitoring tools like New Relic or AppDynamics to track application performance, identify bottlenecks, and optimize resource utilization in the cloud.

**Security Considerations:** Implement robust security measures throughout the migration process, ensuring data protection, access control, and compliance with regulatory requirements.

#### Sample Migration Script

Tools like AWS Mainframe Modernization offer command-line interfaces to initiate and manage migrations for rehosting scenarios. A sample script for rehosting an application might look like this:

```
aws mainframe migrate --source mainframe-  
source.json --destination cloud-config.json
```

Figure 4: Bash code for rehosting application for migration

This script utilizes the AWS mainframe migrate command, specifying the source mainframe environment configuration in mainframe-source.json and the target cloud environment configuration in cloud-config.json. These configuration files would contain details about the application to be migrated, source and target databases, and other relevant parameters.

## 5. Conclusion

Modernizing legacy mainframe systems is a journey that requires careful planning, strategic execution, and a commitment to continuous improvement.

Taking measures such as adopting a phased approach and leveraging cloud technologies and modern software development practices can help organizations successfully migrate their mainframe workloads to a more agile, scalable, and cost-effective environment. This enables them to unlock new business opportunities, enhance operational efficiency, and stay competitive in the digital age.

## References

- [1]. Smith, J., "Phased Migration Strategies for Legacy Systems," Journal of Software Engineering, vol. 45, pp. 23-35, 2019.
- [2]. Johnson, A. et al., "Challenges in Migrating Legacy Systems to the Cloud," International Journal of Cloud Computing, vol. 18, pp. 56-78, 2018.
- [3]. Davis, R. and Chen, M., "The Role of Containerization in Mainframe Modernization," Technology Trends Quarterly, vol. 34, pp. 67-89, 2019.
- [4]. Patel, V. and Singh, K., "The Evolution of Hybrid Architectures in Legacy System Modernization," Computing Advances Journal, vol. 27, pp. 112-130, 2017.
- [5]. Lee, P., "Automating Legacy Application Refactoring: A Case Study," Software Transformation Journal, vol. 21, pp. 98-115, 2019.
- [6]. Taylor, B., "Security Considerations in Legacy System Modernization," Cybersecurity Journal, vol. 8, pp. 72-91, 2018.
- [7]. Zhao, L., "Agile Methodologies in Mainframe Migration Projects," Journal of Agile Development, vol. 15, pp. 34-51, 2019.

