



MODERNSYSTEMS

WHITEPAPER

MODERNIZATION SECRETS OF THE FORTUNE 1000

HOW THREE GLOBAL BRANDS USED UNCONVENTIONAL MEANS TO
TRANSITION LEGACY SYSTEMS

Introduction

Legacy systems have been in place for decades. Most are based on aging development standards with limited functionality and usability. Despite their limitations, many are still relied on to provide mission-critical business functionality. Conversely, it takes a flexible, nimble organization to compete in today's business landscape. This paper highlights the drivers to modernize, alternative choices to common solutions, and profiles three global businesses who leveraged these alternatives for success.

Who Should Read This: Relevance by Organizational Role

This content specifically helps:

- **CIOs** looking to streamline systems and add efficiencies
- **CFOs** looking to dramatically reduce license spend associated with legacy systems
- **Line of Business Managers** looking to integrate platforms for better service levels and performance
- **Business Analysts** seeking to leverage data analytics and/or business intelligence to gain competitive advantage
- **Technical Managers** seeking to end dependency on a legacy technology

What You'll Gain From Reading This Document

This document incorporates industry analysis, direct customer feedback and 20 years of project experience to help you **understand, build and execute** on a business case for modernizing legacy environments. The content within:

- Summarizes the modernization business case
- Shares three alternative paths for modernization
- Shares three success stories of global brands that leveraged these alternative paths
- Identifies Best Practices and Critical Risk Factors for project success

Relevant Environments and Technologies

- z/OS, z/VSE, OpenVMS, ICL VME
- Applications: COBOL, Natural, CA GEN, PL/I, ADS/O
- Databases: VSAM, CA-IDMS, IDMS-X, Adabas

About Modern Systems

A global leader in modernization since 1983, we have:

- Been trusted by Walmart to modernize the world's biggest order processing system
- Completed several large scale, successful modernization projects worldwide
- Developed a solution that offers the most choice and least risk for modernizing legacy applications and databases

The Modernization Business Case

With business buyers and IT buyers both striving to extend the life of existing applications, yet still bring additional functionality and value to the business, modernization efforts are being driven by specific business outcomes.

There has been an increasing focus on business operations, risk and growth driving modernization rather than simply cost reduction. The focus on business outcomes enables IT and the business to focus investments where the business outcomes are richest and also prioritize initiatives from a portfolio strategy. The following are typical drivers used by business and IT managers to trigger modernization efforts.

The Challenge of Legacy Systems

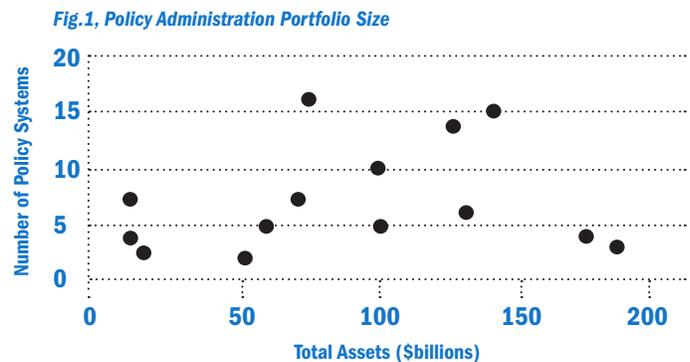
Most legacy systems have been in place for decades. Business challenges perpetuated by these legacy systems can include:

- Disjointed development and maintenance, leading to “silo-based” usage
- Maintenance and staffing costs are high and growing
- Compliance and regulatory challenges
- Limited functionality and integration of legacy systems limits innovation and growth
- Architecture makes business intelligence efforts either impossible or prohibitive

Because legacy applications were built in a different time, often to serve a single purpose, they rarely meet the criteria necessary to help an organization for the long-term. It’s important to understand the specific impact of legacy systems on competitive advantage and business growth.

Repetitive, Disconnected Systems

In some cases, similar functionality exists off the legacy stack, creating dual maintenance scenarios. According to benchmark data collected by CSC (Figure 1), a typical mid-to-large U.S insurer maintains four or more policy admin systems, and many have portfolios of 10 or more.



Sources: AM Best 2011 and CSC Sample of 14 US insurance companies

Some multinationals have redundant platforms across countries or regions, owing to earlier growth through acquisition strategies and subsequent governance of each region as a standalone profit center.

Organizations reap the biggest returns when they are able to reuse data and avoid redoing or completely recreating work that’s already been done. Without adequate interfaces linking systems, reusability is greatly limited, if possible at all.

Legacy applications simply were not designed for this. With their unique and hard-coded interfaces written with rigid functionality in mind, legacy systems make integration difficult. This further perpetuates existing silos and prohibits sharing of information across departments and line-of-business (LOB) applications.

The Modernization Business Case

Disparate Systems Destroy Advantage

Certain industries depend on interconnected data to maintain efficiencies and grow competitive advantage. For example, a retailer may first record customer information with a marketing initiative, but want to reuse this information to offer a personalized credit card or affiliate special. If you work in an industry with these types of interconnection requirements, reusability of data is likely a top priority.

Without major time and dollar investments (not to mention serious risk) to current systems, integration and reusability may be impossible for companies wishing to remain in a legacy state. Programmers spend so much time maintaining and rewriting legacy code in an attempt to keep up with user demands that they have little if any time to create new or unique functionality to improve business processes.

On the business side, with their hard-to-learn interfaces and lack of functionality, legacy systems hardly facilitate productivity. The outcome is that business users often spend more time trying to decipher the system or actually avoid using the system at all, opting instead to complete tasks manually.

Not only does this cost time and increase employee frustration (which leads to staffing and morale concerns on the business side as well), but when employees revert to manual processes, information security and access become crucial concerns.

Compliance Quicksand

The Institute of Internal Auditors linked obligations under the Sarbanes-Oxley Act to a variety of specific recommended IT practices, including the upgrade of operating systems and other software “to stay current with security patches and to ensure continuous support for all software in use.” Failure to patch systems promptly or continued use of unsupported software can be viewed by courts as negligence.

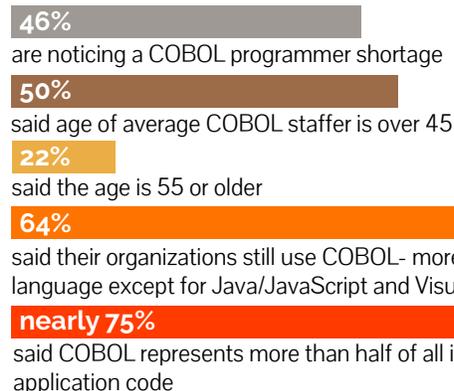
In many cases, legacy systems have been updated by many hands over the years with relatively minimal testing and release note documentation. Customization of these systems can easily generate an operating deficiency – in SOX terms, this occurs when a control is not operating at it should—or there might be a technical fault present in the system. If any one of these is directly involved with the administration of a company’s finances and official support is no longer available, then the company is at risk of a serious SOX violation and C-level management must bear the legal consequences by default.

Who Will Help You Climb Out?

Today’s technology professionals are focused on modern languages, like Java or C#, taught in rich detail at universities and reinforced globally through online user communities.

According to research, the largest volume of skilled developers introduced to the job market in 2013 was Java programmers (32%), followed by C# and C++ programmers (16%). In comparison, only 5% were COBOL developers.

Another 2013 survey from ComputerWorld IT further reinforces the reality of the legacy skill shortage:





Modernization Secrets of the Fortune 1000

While an aging pool of knowledgeable developers, high maintenance costs, and limited system flexibility are forcing businesses to re-think their resistance to modernizing their legacy environments, there is still a great deal of reluctance to do so. However, new, lower risk options for modernization are now available. They reduce cost and enable the business while avoiding the time and resource burden that firms typically associate with a rip and replace modernization effort.

Isolating Workloads to Reduce Cost

One reason legacy mainframe systems are still relevant is that they're astonishingly adept at processing extremely large amounts of data with exceptional performance and reliability. The volume of data processed is measured in MIPS (Million instructions per second); the higher the number of MIPS, the greater the cost.

Many organizations will target MIPS reduction to reduce mainframe operating costs, thereby freeing up resources to invest in other key IT initiatives.

Offloading workloads to reduce MIPS is relatively easy and transparent to the user experience when planned and executed properly with adequate computing resources. The exercise most often consists of converting one or more batch process applications to a modern programming language such as Java or C#.

Organizations who offload batch processes in this manner tend to allocate the cost savings they realize to further the overall system's modernization effort.

This practice is common because enabling the processing of workloads out of the mainframe environment consists of the conversion or re-engineering of a subset of the legacy code, essentially taking the first steps towards the overarching modernization effort.

Focusing On Data

The practice of using business intelligence (BI) systems that manipulate and present this data in meaningful ways equips corporate decision makers with the information they need to maintain and improve business performance, successfully scale as their businesses grow, and identify and implement new business initiatives and strategies.

However, problems arise when trying to integrate data from the non-relational mainframe environment with business intelligence systems that are ubiquitously built to interact with relational databases. The result is data lock-in.

The liberation of legacy data through conversion to a relational model is a common driver for modernizations. However, many firms feel as though a complete re-engineering effort encompassing the entire legacy system is too risky and time-consuming.

Luckily, a less stressful option exists for liberating this data while keeping the legacy system at-large intact. Data replication allows a firm to replicate the non-relational data from their legacy environment into a new environment expressed in a relational model. The legacy system works exactly as it did before, the user interface and underpinnings are the same, as is the database. However, the data is copied and converted to the relational model automatically, then refreshed on a schedule, providing an up-to-date replica of the legacy database in a form that BI systems can understand.

In essence, data replication allows the organization to maintain their existing legacy environment, while liberating the data into a relational model to provide the visibility and flexibility that the business wants.





Modernization Secrets of the Fortune 1000

Automated Conversion

While batch offloading and data liberation through replication solve many of the issues companies face with respect to legacy systems, the business remains at great risk.

Many companies are running mission-critical applications on legacy systems that do not have a development roadmap that accurately reflects the value of the data stored on the platform, hindering the ability of developers to sufficiently augment legacy systems to meet the demands of the modern marketplace.

Automated conversion, harnessing the proper tools and expertise, produces like-for-like functionality from the legacy system to the new environment utilizing fully maintainable modern code such as Java or C#. This means once the legacy application and database is translated, developers can extend application functionality directly. Precious business logic from the legacy system is preserved while enabling deeper integration and customization to meet business requirements. It also opens up a whole new world of scalable resources for system support. Since conversions are software-based, costs are significantly lower than those associated with re-engineering efforts. In addition, automated conversion projects finish 75% faster than rewrites.

Automated conversion is particularly attractive because unlike re-engineering efforts, it can be accomplished in steps. Since database conversion and batch offloading are both components of an automated modernization, an organization can ease into the effort by liberating the database into a relational model, offloading batch applications, and completing the remaining legacy system's modernization through automated conversion.

When To Consider These Alternatives

Karen Furtado, Partner at Strategy Meets Action says the key is to approach the modernization in a strategic and planned manner rather than as a tactical project. The following are the common options for modernizing systems.

Known as the “big bang approach”, re-engineering is the most expensive and risky solution for modernization. It includes requirement capturing, coding, debugging, testing and refining. To recreate the legacy solution's wealth of functionality, with newly written applications requires significant time and effort. Impact on end users and their adjustment cycle is also a major factor. Many analysts consider the failure rate of these projects to fall between 70-80%. The cost, time and risk involved with re-engineering are so high, few organizations choose to embark on large-scale rewrite projects.

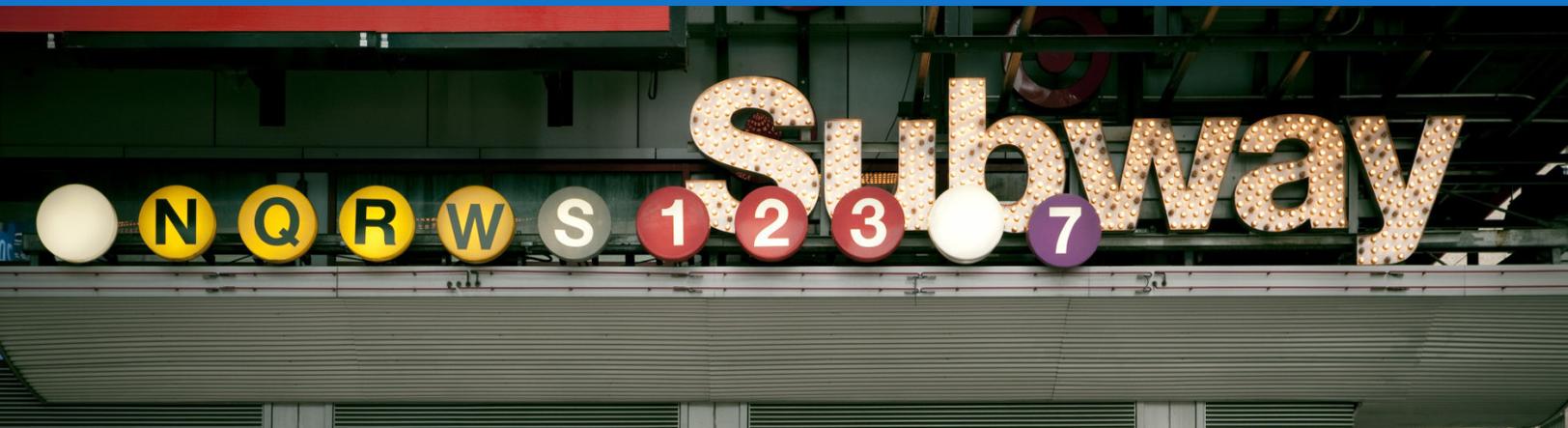
“Off The Shelf” solutions focus on replacing legacy application functionality with packages and components available from third party vendors. The positives of this approach include reduced maintenance of source code, as the vendor is responsible for fixing production bugs and implementing new functional enhancements.

However, commercial packages offer standard domain business processes that often differ from the homegrown legacy application. Reuse of existing business logic isn't possible, so some level of business process re-engineering or customization/rewriting of the third party solution is required. Both of these processes can be time consuming and expensive.

If your company is seeking to reduce cost and increase business value of your legacy applications - while minimizing impact on everyday users and IT - it's worthwhile to consider these alternatives.

The following companies, BBVA, Metropolitan Transportation Authority, and Companies House, all evaluated traditional legacy rewrites and off the shelf solutions - but chose alternative paths. Here are their stories.





Batch Offloading: Metropolitan Transportation Authority

Phased Modernization effort starts with batch process applications, cost reduction

Background

The MTA is a New York public authority and public benefit corporation that provides transportation services in the NY Metropolitan region through its operating agencies, namely:

- New York City Transit (subways)
- Long Island Railroad
- Metro-North Railroad
- Bridges and Tunnels
- Staten Island Railway
- Capital Construction Company
- MTA Bus Company

The MTA is running an IBM mainframe. Part of their overall IT strategy for reducing cost and risk includes moving critical applications off of the legacy mainframe environment to newer, more flexible and cost-effective platforms. The first selected for modernization is called IMPACT (Integrated Management of Payment Accounting and Capital Tracking), a batch process application. IMPACT is built in COBOL, leveraging VSAM as the data tier.

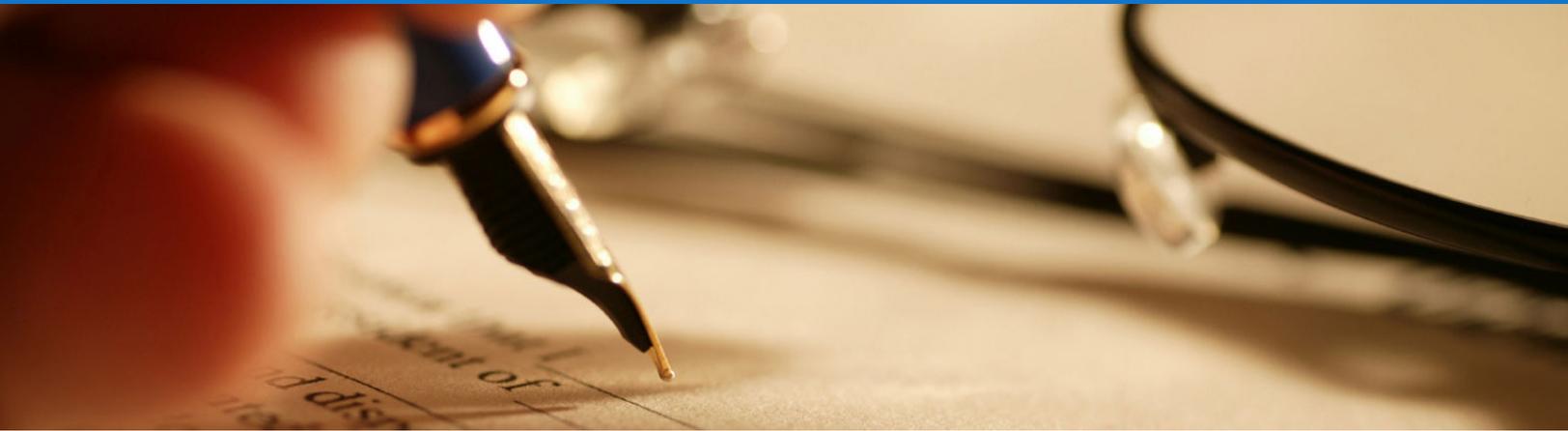
Solution

MTA considered the option to re-host the IMPACT application in a COBOL container on a newer platform, but preferred the alternative of converting the IMPACT application from COBOL to Java for improved future maintainability, flexibility, cost-effectiveness and support. The converted environment enhanced MTA's position to provide improved, cost-effective services to its customers. The data tier was converted from VSAM to Oracle Database.

Going from COBOL to Java isn't a trivial undertaking. Understanding all levels of functionality and relationships is essential for success. Modern Systems started by leveraging our Mainframe Assessment service to assess the application's technical inventory and risk factors. Then, we worked with MTA to finalize test strategies and critical factors to validate equivalent functionality between the source and target systems.

Since our code conversion solution is 100% automated, any errors found during testing are most likely be common to similar code in other parts of the application. This leads to a testing approach which tests all major items at a high level and then some specific items at a much lower level. Making sure the test scripts that are used cover the right scenarios is a vital part of the project's success and problem-free migrated applications.

In the end, MTA's mainframe application portfolio runs the same as it always has without any visible or impactful change for users. However, by moving batch processes off the mainframe, MTA will achieve its cost-savings goals and begin to validate their organizational approach for modernization.



Data Replication with Companies House

Portfolio Optimization With Mainframe DataShare makes data available without risk of rewriting apps

Background

Companies House is Britain's government bureau with responsibility for recording and storing all information that the country's 1.8 million limited companies must share with the public. For years Companies House gathered this information in their customized IDMS application. They elected to move this system to a Unix platform running Oracle in order to curb costs, streamline development work on the application, and simplify user access to data via the web.

A key challenge the organization faced early in this initiative was how to move all of the data from IDMS to their new relational database. Companies House sought a product that could accurately translate the IDMS constructs and drop the data into the right tables.

Modern Systems' solution could rebuild their hierarchical database in whatever relational model they chose (Oracle, DB2, SQL Server). Then, it would load the data quickly into the new target database.

Solution

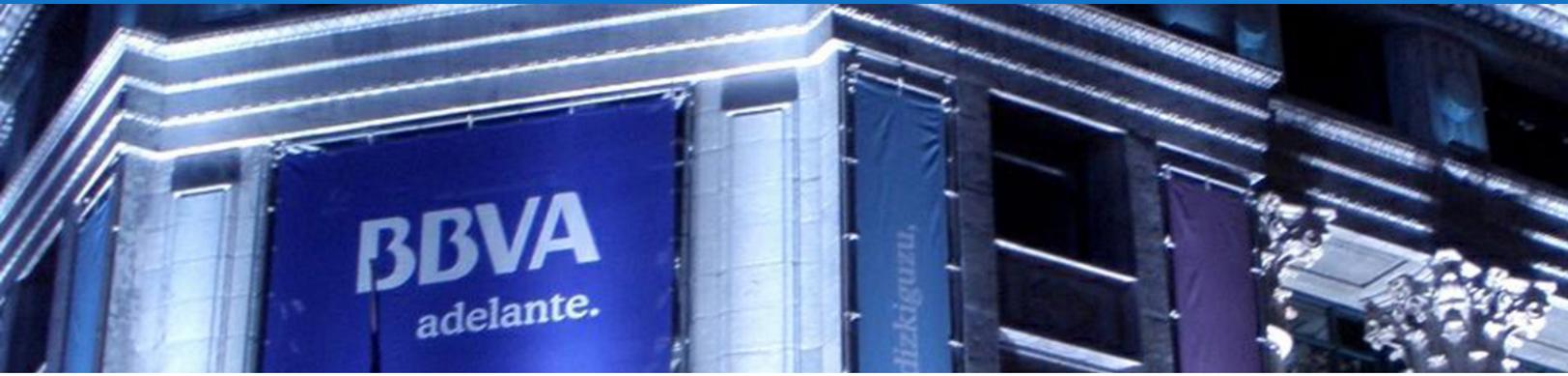
Companies House realized that this technology would save significant time building an Oracle database that was perfectly consistent with their existing IDMS database. Furthermore, Carl Allen, Technical Architect for Companies House, explained that, "Modern Systems could give us a continuous refresh of the data right out of the box." Daily data updates are important to Companies House because of the need for public access to newly added registrations and filings.

First the IDMS schema, subschema, records, and other database components from the IDMS Data Dictionary were exported from the mainframe to the Mainframe DataShare System Dictionary on a staging platform. Since all of the conversion work was then performed on this staging platform, and did not impact the mainframe, Companies House was able to continue with its normal end user and IT activities throughout the project.

Once these components were in the dictionary, Modern Systems' powerful mapping language converted the data model in three basic steps. First, it translated Companies House's IDMS Records, Elements, and Sets to relational tables, columns, and foreign key constraints. Second, it took the IDMS data constructs and created the appropriate relational data types. Third, it generated all of the necessary XML, XSLT scripts, and SQL scripts which then built the target Oracle database.

Modern Systems provided Companies House with a viable, secure way to convert its critical business data from a 20 year old IDMS system to a modern relational database. In addition, Modern Systems made it simple to keep the database up to date with the newly arriving information. The new system resulted in meaningful financial benefits for Companies House and faster service for its customers.





Automated Conversion: BBVA

Drivers include cost reduction, data integration, and feature set enhancement

Background

As many a Gartner analyst will attest, legacy modernization is becoming a high priority for banks across the world. The ability to reduce costs while enabling data and business process integration is necessary for competing in today's global banking environment.

BBVA is the second largest bank in Spain. Since 2006 the bank has focused on overseas expansion, and now operates in 40 countries. Like many other Spanish companies, it enjoys a top tier position in Spanish-speaking Latin American countries. BBVA's debt market operation in Mexico, Chile, Colombia and Uruguay is dominant. It has a strong retail presence in southern European countries, especially Spain, Portugal and Italy, and has expanded into the United States and Asia.

VASS Consultoria de Sistemas – one of the major consulting companies in Spain – was selected to run the project as the prime contractor. Modern Systems was subcontracted to perform automated conversion of IDMS, ADS, COBOL, Easytrieve and JCL. The converted systems went into production in September 2013.

Solution

BBVA's primary objectives were to cut licensing costs of the mainframe and free up data trapped in an IDMS database. BBVA decided to migrate off the mainframe two of its applications – International and Renting – both written in COBOL, ADS, and Easytrieve. The target for the migration included Java and Oracle database. BBVA wanted the new system to be an exact copy of old system, as the legacy application's business logic and user interface were satisfactory to the business.

As with most modernization projects, deep understanding of the source system is essential. This is often difficult, as these systems can be over 30 years old, having passed through multiple hands with minimal documentation. A Mainframe Assessment exercise is the first step in every Modern Systems project. This maps out the total technical inventory, technical use cases, identifies risk factors, and unused programs. For the COBOL applications, we identified large amounts of dead code and 876 unused programs out of 2311 supplied programs – approximately 38% unused programs!

Upon successful launch of the online and batch applications, overall system costs were cut significantly. Just considering direct license savings, BBVA project leaders predicted positive ROI for the entire effort would be achieved within 2 years. There was the additional strategic benefit of risk reduction around IDMS resources, which are expensive and difficult to find. A big positive of legacy modernization is the ability to leverage highly available, cost-effective resources to run the target environment.

Executing The Plan: Best Practices and Risk Factors

Understanding the details of an application and its interrelationships with other systems is a necessary step for modernization. A challenge for many companies is that little documentation exists, or is current, for many applications. The tendency is to move forward without it.

You Don't Know What You Don't Know

Obtaining good code understanding and creating documentation helps organizations make better modernization decisions. Companies should not simply take a symmetric view of the transition. Some parts of the portfolio may be moved with little change. For others, the organization may have to evaluate alternative ways to provide the desired function.

For example, replacing Natural applications that create reports with report-writing packages eliminates the need for code migration or to rewrite the application. Understanding the specifics of these applications enables an organization to plan the conversion to a report-writing package.

However, understanding the business rules in the code is necessary to support the modernization initiative. For this code understanding and documentation step, leverage an automated discovery and documentation service like Modern Systems' Mainframe Assessment to quickly and accurately understand your Natural/Adabas ecosystem.

Understand Operational Consequence

Users should experience a "Business As Usual" feeling throughout and after the completion of your project. Any operational impact slows project delivery and creates negative perceptions internally about the project's viability. Maintain an on-going data conversion strategy that allows new features to be built while the system is modernized.

Hitting Your Target

The information obtained during a Mainframe Assessment helps define the complete destination architecture. Organizations may have a predilection toward Java in a Unix or Linux environment, or a Microsoft .NET strategy on Windows. The definition of destination architecture, in line with the organization's strategic enterprise architecture strategy, and a subsequent mapping of the old applications and operational issues to this environment, helps identify gaps in capability.

Performance differences are important considerations. Procedural porting may be less elegant architecturally, but more likely to meet performance needs. Process redesign may be necessary to provide the same apparent user response, despite lower absolute performance levels.

Don't Be Afraid To Ask Hard Questions

As part of this planning step, organizations should ask several questions as they evaluate a migration effort:

- What are the implied aspects of total cost of ownership (TCO) and platform selection, such as backup, recovery, security and availability?
- How can we gracefully migrate, while minimizing the effect on current software contracts?
- What are the effects of such a migration on staffing and skill levels?
- What quality of service can be achieved, and will it be sufficient?
- How long will it take to build the same operational skill levels?
- What are the implications of migrations on record retention and audit implications?



Organizations should consider the difficulty of extricating themselves from potentially decades of architectural dependence. Although many technical differences affect such projects, the impact of change on people and the processes they use to deliver and operate applications is significant. Follow these Best Practices to minimize risk of your transition.

Executive Buy-In

Understanding the reach, risk and scope of modernizing a legacy platform can involve cross-functional resources. It's important for executives to understand and support the proper use and prioritization of tasks for those impacted by the project. Without executive buy-in, the likelihood of your project falling behind anticipated timelines skyrockets.

Proactive, Participative Management

Mobilize stakeholders during the entire project by setting up interactive means of communication (management committees, newsletters, master plans) as well as during the transition period. Consider rigorous management of the following activities:

- Management committees encouraging synchronization, convergence and communication
- Structured accountability at every decision-making level
- Formal information mechanisms (newsletters, minutes from meetings, progress reports during by project team meetings)
- Recognition of successes by IT resources

This level of activity is often necessary to ensure change strategies (communication, transition and training plans) are implemented according to plan.

Strategic Evaluation Of Legacy Functionality

Proper resources and timelines must be applied to assess the project's impact across the business. Legacy artifacts and logic should be measured for relevance to future business goals and requirements. Business logic and processes relevant to the business should be added to the project plan. Those no longer needed should be refined or removed.

Proof-Driven Vendor Selection

A substantial Proof of Concept is necessary to validate vendor technology. This process also identifies aspects of the project that require internal customer resources.

Commitment To Timeline

Change requests during the late stages of your initiative could significantly delay or even derail the project. Escalation management procedures ensure alignment around critical issues, avoiding ambiguity and misalignment during the various implementation phases.

Thoughtful, Exhaustive Testing

Cross-departmental representation is essential in developing test cases/scenarios for integration and system testing. Start testing early and involve end users. This reduces risk by identifying and solving unexpected integration challenges, underestimated efforts and undetected functional gaps.

Commitment to Performance

Prioritize performance with attention to redundancy, recovery and database availability. This focus enables implementation of agile software development and creates opportunities to deliver new, value-added services to the business.



Conclusion: Modernize Without Compromise

Transitioning away from legacy systems is a complex undertaking. It's different from application to application, and business to business. Modern Systems provides a flexible, incremental approach that balances cost, risk, and time in a way that meets each customer's unique requirements.