OpenLegacy

14 Essential Elements of a Successful Legacy Modernization Process

White Paper

14 Essential Elements of a Successful Legacy Modernization Process

It is easy for any IT professional to get distracted by technology. Vendors, analysts and reporters are constantly trumpeting new technologies and even mainstream television advertisers are trying to convince the world that their approach to data management or application development is the only way to succeed in today's competitive environment. All this talk about cutting edge solutions masks an important truth: Information Technology has progressed so far that now mainframes and legacy data sources can co-exist harmoniously with next-gen mobile, web and cloud solutions.

While modernization no longer requires a cumbersome hardware upgrade, there are certain key elements any strategy needs in order to be successful:

01. Focus on creating value

Modernization isn't about technology for the sake of technology. Instead, it's about using Information Technology to solve real business problems – and then finding the fastest, most efficient ways to achieve it. Rather than focusing on server speeds or query response rates, IT groups need to start with an understanding of key corporate goals and ideal business processes. Today the modernization effort can begin with solutions that offer customers new mobile services or specialized workflows that let call center agents quickly answer specific requests – not an architectural blueprint.

02. Decouple hardware from software

Recent advances in IT – virtualization, IaaS, software-defined networks – all stem from the efficiencies gained when network and IT functions are controlled via software and separated from the underlying hardware infrastructure.

The same principles apply to legacy modernization projects. Companies can now ignore hardware upgrade issues and focus instead on developing custom applications that use a mobile or web-based interface to read/write data to legacy sources. These solutions can even pull data from multiple systems if needed as functionality is dictated by business requirements, not IT constructs. Taking a software-based approach to modernization is essential for creating a cost-effective, value-focused program.

03. Leverage open-source programming methods

While many believe that legacy systems and standards-based solutions are incompatible, nothing could be further from the truth. The advances in open systems actually enabled this paradigm shift in legacy modernization, one that leverages software, APIs and standard server technology rather than middleware, complex integration schemes and risky hardware upgrades to achieve corporate goals.

It is essential to any modernization program that all application development work be done using open-source programming languages. From a project perspective, open-source programming languages such as Java are extremely lightweight. In fact, they require 80% less coding work than COBOL and other legacy languages. More importantly, standards-based solutions provide greater long term flexibility and portability options. Over time, adhering to open, standards-based methodologies will limit future recoding/reintegration work and ensure that corporate growth will never again be hampered by vendor lock-in and/or IT constraints.

04. Embrace APIs

Previously, legacy systems were forced to utilize complex middleware and integration schematics to enable communications with web solutions, end user devices and third-party applications. In most cases, these projects were very costly and time consuming. While this approach did allow mainframes and legacy data stores to communicate with next-gen enterprise applications, they rarely proved robust enough to satisfy mobile, web and cloud-based requirements.

Luckily, IT has evolved to a point where modernization efforts no longer require custom integration work or a designated “communication layer”. Vendors and IT groups worldwide rallied around APIs, interfaces designed to provide a standard set of protocols that would enable communication between application code, data sources, plug-ins, servers, devices - all the varied elements needed to enable in a specific solution.

APIs are the cornerstone of any modernization program. Building a set of APIs that correspond to specific legacy data elements allows developers to continuously create a series of modernized applications able to serve a wide range of business needs.

05. Start small – plan big

Because current modernization projects no longer require large-scale, high-risk replatforming work, companies can now take the process one step at a time. After all, simply writing application code that does not alter the data source itself is significantly less difficult than replacing the core of a company’s infrastructure.

Starting with a specific project – restructuring call center workflows or improving the responsiveness of online solutions – allows IT departments to begin the modernization process without overcommitting budgetary or workforce resources in the process. It’s the perfect opportunity to experiment with different application development tools and techniques without incurring a lot of risk. With the first project under their belt, IT groups can proceed intelligently with a wide range of initiatives for optimal results.

06. Enhance customer services

Many organization must modernize to better meet evolving customer needs. Until now, it has been difficult to deliver cutting-edge, highly responsive web and mobile apps that must communicate with mainframes and legacy data stores. Middleware, browser incompatibilities and other integration issues tended to degrade performance. Even the most basic web-enabled legacy solutions have been plagued with delays and customer frustration. The complexities inherent in outdated communication methods tended to leave more sophisticated projects sidelined and/or cancelled outright.

Today, mobile and web apps can communicate directly with legacy systems via APIs as efficiently as any server-based solution can. This is critical to any modernization program. By removing the largest go-to-market barrier companies previously encountered when trying to launch online services, IT groups can quickly and easily deploy the innovative solutions that drive retention rates and revenue generation.

07. Increase employee productivity

Over time, the rigid requirements imposed by AS/400 and mainframe screen navigation has made virtually all legacy applications cumbersome and inefficient when they are compared to evolving business requirements. Modernized applications can eliminate these issues. Although they are tied to information located in legacy systems, their application logic is not. Developers can easily revamp these solutions with input/output forms that better match current workflows. More importantly, they can enrich applications by uniting information from multiple sources into a single solution. Internally oriented modernization solutions that cut the time it takes to access and update specific information sets will generate significant productivity savings. Additionally, they can help seed department-specific activities – reducing sales cycles, streamlining logistics – and enhance decision-making in the process.

08. Serve the mobile workforce

Every IT department struggles with BYOD issues and finding ways to effectively deploy and support apps across so many different smartphones, tablets and laptops. Addressing these issues within the modernization process from the start will help extend project longevity. As with any other mobile or web app project, developers will need to tweak the apps for target screen size and hardware profiles. Development tools that offer a built-in design studio will help users mitigate form factor issues without having to worry about how those changes will affect other devices.

09. Prepare for the cloud

To decrease CAPEX, OPEX and the daily burden on IT, an increasing number of infrastructure, network services and business applications are being migrated to the cloud. Even those companies without immediate plans to jump should ensure the ability to extend applications to the cloud within their modernization strategy. Without this level of forethought, companies may find themselves unable to work with key partner, customer and supplier solutions. Incorporating open-source development techniques capable of exposing RESTful APIs into the modernization process is critical to ensuring compatibility with cloud-based environments.

10. Optimize user interface

Creating a compelling UI is critical for driving higher adoption and engagement rates for both internally and externally-facing applications. Therefore, design should be considered critical to any modernization project. Choosing a solution with style sheets and templates allows developers to quickly create visual interest. Incorporating third-party plug-ins, widgets, breadcrumbs and more into mobile and web apps goes a long way towards building familiarity and encouraging usage.

Modernization solutions that support a flexible API tree and data type information allow developers to further enhance usability with pop-ups, specialized forms and streamlined workflows.

11. Ensure application responsiveness

Users now expect lightning fast response for any mobile or web application they use. Delays of any kind will hinder adoption, productivity and customer satisfaction rates.

To be successful, modernized solutions must be able to support millisecond communication rates with back-office systems. Streamlining integration via APIs and implementing connection pooling will help ensure that these solution meet the stringent performance specifications needed to satisfy internal and external users alike.

12. Enable agile, devtest processes

Just because project involve legacy hardware does not mean that IT cannot use agility to speed modernization itself. While agile processes and tools may vary from company to company, the benefits are universal – shorter development cycles, faster issue resolution and more robust solutions.

Dovetailing modernization work with existing IDE software, CI procedures and Java development tools will enable companies to leverage existing skill sets and agile processes. Choosing a modernization platform with built-in testing tools will ensure that companies detect and correct bugs early in the development process when they are less costly to fix.

13. Leverage automation tools where possible

Nowadays, there is little glory in grunt work – instead IT groups are measured on ROI, time-to-value and other business-related KPIs. Unfortunately, there can be quite a bit of drudgery in any modernization effort. For example, creating and specifying the numerous APIs needed to tag each mainframe screen and data field would be extremely time consuming and tedious if done manually. Once the backend communication protocols are established, coding the application logic, designing for multiple device types and optimizing responsiveness can also be time consuming without the proper shortcuts and resources.

Compared to the cost of replatforming, investing in tools that speed modernization work itself are minimal.

14. Lower the Total Cost of Ownership

No modernization solution can be considered successful if it doesn't generate ROI quickly or consumes the company's entire IT budget. Modernization techniques that focus on software development – as opposed to buying new hardware – enable organizations to increase the value IT generates with a minimal investment.

Typically, software-based modernization techniques will reduce TCO by 75%.

The Bottom Line

Legacy modernization should no longer be considered a risky, expensive undertaking that requires a company to dispose of key infrastructure investments. With a strong, flexible foundation for application development, companies can quickly, confidently modernize legacy investment in a way that is financially viable. More importantly, at the end of these projects companies will gain real value for the company as a whole – not just a costly set of new hardware or unused vendor software.

OpenLegacy
103 Carnegie Center, Suite 300
Princeton, NJ 08540
609-608-0556
info@openlegacy.com

Main Web Site
http://openlegacy.com

Developer Zone
http://developer.openlegacy.com