

Strategy for Application Modernization

A Summa White Paper



summa
Technology + Business

Summa
925 Liberty Avenue, 6th Floor
Pittsburgh, PA 15222

(p) 412.258.3300
(f) 412.258.3299
www.summa-tech.com

Why Modernize?

“My customers want up to the minute information about their accounts, but we post nightly. I can’t afford to rewrite everything. What should I do?”

“This is our third merger in five years and I can’t continue to cobble together these applications, but I can’t totally replace them either. What are my choices?”

“I finally got the letter from my vendor. They won’t support my system beyond this year. Where do I start?”

“My applications run fine, but it’s getting harder and harder to find the right people to support them. It’s time to start transitioning to a new environment. Who can help me with this?”

“Our auditors are breathing down our necks about exposing customers’ personal information in our legacy system. What can we do?”

Application Modernization

According to Gartner, by 2010 more than a third of IT projects will be driven by the need to modernize their aging IT assets and update the skill sets of their employees. In many cases, this technology obsolescence has been compounded by the lack of integrated planning and coordination of critical IT assets within the IT and business management teams. Companies need to develop the right application modernization strategy so they can continue to operate efficiently while retaining the flexibility to adapt to the demands of a dynamic market.

Successful application modernization projects will create an architecture that allows your applications to evolve as your business needs change. These application modernization projects typically involve creating new business value from existing applications, incrementally transforming legacy systems into new reusable business components, or leveraging existing enterprise skills and improving productivity. Businesses must continually balance the current state of their IT infrastructure with market drivers, business drivers and IT drivers depicted in Figure 1.

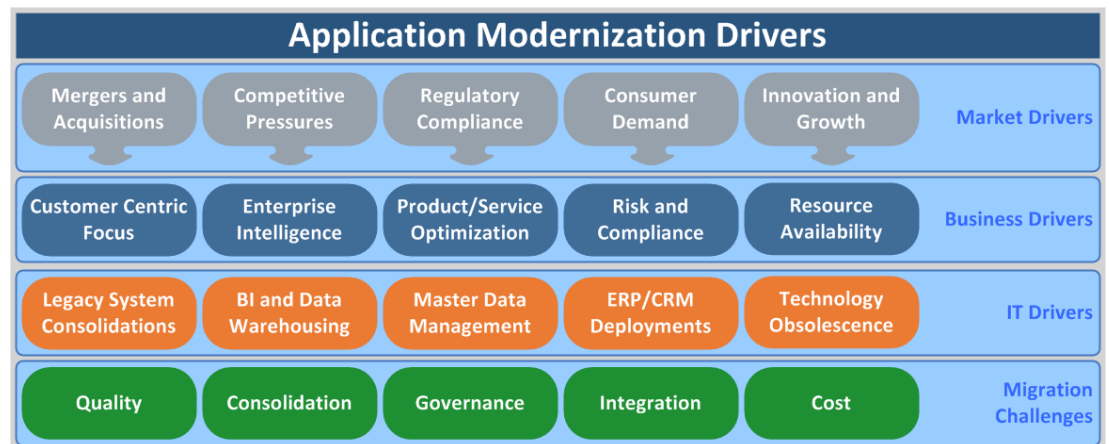


Figure 1

This paper discusses the primary factors that drive an organization’s need to modernize applications and it will review how to approach a modernization project by establishing and executing a realistic strategic roadmap that starts from the current state environment, captures existing and new business and IT goals, budgets and skills, establishes the future strategic architecture and charts a path through a custom strategic transition roadmap.

Why Modernize?

What signs should you look for to indicate that your application is in need of modernization? Start with the factors that are causing a drag on your business such as staff inefficiencies, processing errors, lost customers, inaccurate data, broken business processes, regulatory compliance warnings, or high IT maintenance costs.

Some of the common reasons driving an organization's need for an application modernization project include the following:

Market related drivers

- Merger, acquisition or divestiture driven integration.
- Regulatory compliance requirements from the SEC, Sarbanes Oxley, or Basel II.
- Competitive and customer pressures to add function, performance and capacity.
- Business innovation opportunities.

Business drivers

- Need to better integrate with other business systems: internal, supplier, or customer.
- Increased customer focus.
- Time-to-market deadlines.
- Budget constraints.

IT drivers

- Lengthy development cycles.
- Product and technology obsolescence and ending vendor product support.
- Lack of IT skills for application maintenance.
- High cost of vendor lock-in and maintenance.
- Poor performance, reliability, or availability.
- Security audit compliance and access issues.

Organizations that develop a strategic application modernization architecture will realize some of the business benefits of modernizing including:

- Business agility to rapidly adapt to continually changing business requirements.
- Improved customer service.
- Tighter integration with partners and suppliers.
- Ability to exploit off-the-shelf products.
- Reduced total cost of ownership.
- Improved data quality.
- Improved security management and control.
- Improved operational efficiency.
- Faster time-to-market

Establishing the Modernization Roadmap

A modernization roadmap establishes consensus for the alignment and vision around the strategy, pace, budget and timeline between lines of business executives, business partners, IT application development staff and IT operations staff. Modernizing a critical application typically presents significant challenges that impact many parts of a company. So where do you start? What are the short, medium and long term goals and milestones? How are key plan activities related?

To form a strategic modernization roadmap, start with an assessment of the current state of the IT environment and a desired future state vision. In addition to IT architectural considerations, it is important for the assessment to consider the following factors as input to the roadmap:

- Direct and peripheral business goals, functionality and timelines.
- Customer and end-user impact (both positive and negative).
- Dependencies between applications, infrastructure components, partners and service providers.
- Internal and external integration points.
- Staffing impact – end users, IT and partners.
- Capital availability and needs during the scope of the transition.
- Organizational risk tolerance.
- Known strengths and weaknesses of the current solution.
- Other concurrent IT and business initiatives that may factor into the modernization effort.

Structured workshops, interviews and questionnaires are tools used to collect stakeholder input for roadmap decisions and priorities. Establishing and maintaining the roadmap should be conducted iteratively with key decision makers to ensure that consensus about priorities and balance with changing business goals are maintained.

Consider the Impact of Business Process Improvement and Strategic Objectives

Businesses constantly change and adapt to the pressures of the market and consumer demand. During application modernization projects, it is wise to continually look for ways to improve and refactor architectural components rather than simply translate them from an old technology to a new one. Sometimes it may be easier or quicker to implement a refactored business process than to replicate an existing process that was designed with different business and technology constraints. Other times, the cost and impact to change process is so wide and costly that the solution must be crafted to minimize change to users and process.

Defining the Modernization Approach

Figure 2 below illustrates the approach to transition from several legacy systems through an integrated transitional architecture to a target reference architecture. In between the starting point and the target architecture, the transitional architecture and implementation plan definition is necessary as “scaffolding” to hold together the existing applications and data while the system, the users, and overall business processes are being transitioned.

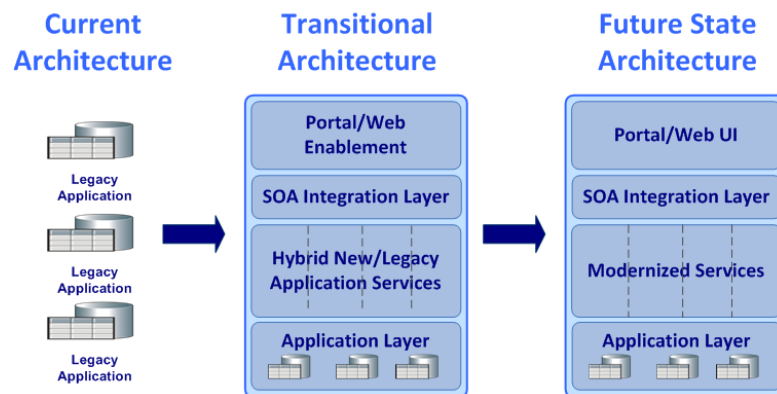


Figure 2

The transitional period can last anywhere from a weekend upgrade where the transitional design primarily consists of data conversion efforts to a multi-year, multi-phase project with numerous components being transitioned independently. Many modernization projects leave legacy systems largely intact, and add web enabled services to allow reuse of existing (and working) business rules and data.

Establishing a Target Reference Architecture

No matter which product vendors you are aligned with, to ensure an agile and scalable business solution, the first step should be to establish a target architecture around a robust and stable application integration strategy. This application integration approach will likely be used both during a transitional implementation and for the final implementation as illustrated in Figure 2.

The foundation of the integration approach should consider the following ideas:

- Base the integration approach on standards-based solutions that support vendor interoperability and independence, broader availability of skills and tools, and a longer lifetime.
- Consider emerging trends such as Software as a Service (SaaS), virtualization, Web 2.0, or open source.
- Establish a loosely coupled, service oriented integration infrastructure – A Service Oriented Architecture (SOA) provides agility for mid-stream adjustment to the modernization effort and long-term agility that allows you to integrate new packaged and custom software as needed.

- Build in architectural capabilities at the foundation - Characteristics such as scalability, reliability, maintainability, availability, extensibility and security are expensive or infeasible to add in later.
- Establish a common IT asset governance model over integration practices – This ensures reusability of integration processes and best practices within and across applications. Myopia during the development of individual applications or services is one of the biggest impediments and hidden cost factors in modernization and integration efforts.

Some modernization projects only consist of laying a new web interface onto a legacy application or directly sharing data access with another application. While important, these types of projects require less up-front architecture and design.

With more significant enterprise application modernization efforts, achieving the goals outlined in the list above requires substantial up-front architecture, design and planning, along with product and process selection implemented by experienced architects.

To define the reference architecture for your application integration environment, we typically recommend that either you start with an industry analyst or vendor defined reference model and design patterns. The reference model will delegate responsibility of products and components within the IT landscape into slices of cooperating functionality based on the integration strategy.

Alignment with widely established enterprise architecture models will allow separation of application functionality and component reuse for long-lived solutions vs. applications built with broad and hybrid responsibility. Flexibility and agility to create composite applications in the future will be greatly increased.

The IBM WebSphere SOA Reference Architecture is depicted in Figure 3 and an example of a logical IT reference architecture is shown on the following page in Figure 4.

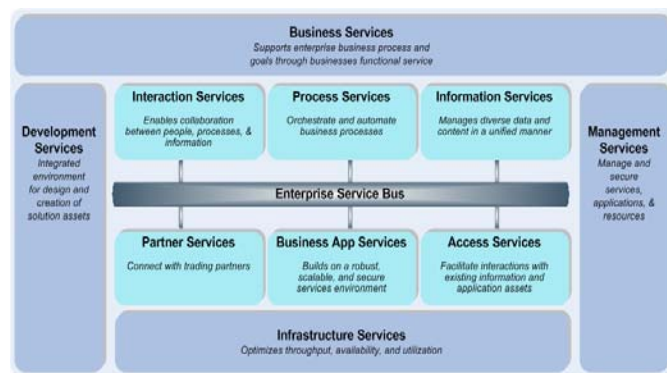


Figure 3

It is also important to consider working your reference architecture into the details of the application architecture if you are developing custom applications vs. integrating packaged applications. Strategies for the implementation of SOA, J2EE, .NET and integration product-specific design patterns and best practices should be adopted to minimize your organization’s need to reinvent the wheel. It is important to keep in mind that the details of the application architecture typically change the fastest over time, have the broadest impact to developers’ productivity and incite the most controversy in the development teams going forward.

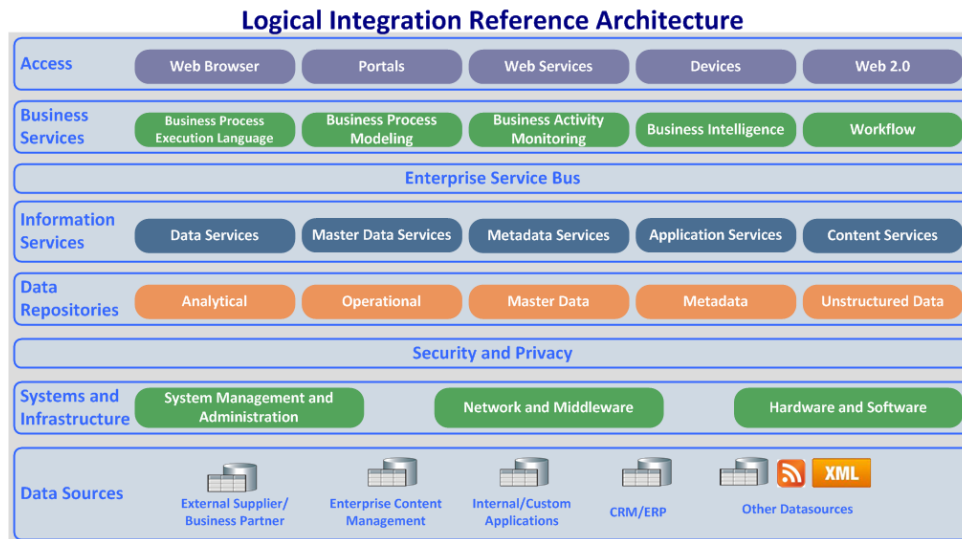


Figure 4

Establishing the Transitional Architecture

After assessing the current state environment and establishing the future state reference architecture, it is time to develop a strategic plan to achieve the desired future state. There are typically many options to consider ranging from “big bang” rearchitecture to incremental migration, each with unique trade-offs to consider. It is critical to take into account all available insight from within and outside of your organization. Difficult choices about the timing of the transition, the phasing of various changes and the speed of transition need to be considered as part of the plan. Careful evaluation of what can and should be “fixed” during the migration vs. what will be maintained also needs to be balanced.

Sometimes a modernization effort must be implemented as a big-bang conversion and other times this is impossible. SOA approaches and portal technologies give us more options today than ever before to more easily create phased implementation plans.

Some of the factors to consider in establishing a transitional architecture include:

- Specify phases to minimize risk and maintain agility as other aspects of the environment are changing.
- Carefully consider ways of dividing up groups of users, customers and functionality vs. the project phasing. Choices about balance between business impact, ease of implementation and need for change should be negotiated diplomatically between IT and business units. Do not forget about details like training users, IT staff and business partners.
- Consider alignment with business objectives and timelines. For example, alignment with yearly or quarterly financial reporting dates or seasonal business activities may be a good or bad thing, depending on the purpose of the application and dependencies that it has on other applications.
- Inclusion of factors beyond the application itself including necessary hardware, operating system updates and network infrastructure or packaged application upgrades.
- Establish realistic risk and contingency plans for various business and technical risks.

To accompany the application modernization strategy, there is a list of transition planning and implementation practices to avoid:

- Short-term thinking – Consider long-term total cost of ownership (TCO) by ensuring that maintainability, scalability and performance are addressed in the earliest project phases.
- Software migration tools that over-promise and under-deliver. Some automated code generation and migration tools greatly increase complexity and difficulty of long-term maintenance, create an over-dependence on vendor's proprietary products, and create a dependency on difficult-to-find specialized skills and reliance on niche vendors.

Another aspect of the transition plan is deciding where to start from the technology perspective. Do you start at the user interface and work your way down through business functionality, architecture and data or vice versa? Newer standardized portal and integration technologies and modernization tools allow more options to cost-effectively build in either direction. Exploring what technologies may be exploited for your environment should be considered in conjunction with creating the roadmap.

Execution of the Modernization Plan

The diagram in Figure 5 on the next page depicts how a modernization project is executed using the RUP® (Rational Unified Process) phases. The project is split into distinct phases that allow management of risks, budgets, skills and investments. Activities are aligned with targeted milestones to first define the vision of the modernization solution, then to establish the infrastructure and detailed plan to support the construction and deployment. This effort is followed by a phased implementation of initiatives to modernize, test and deploy address applications and/or their components.

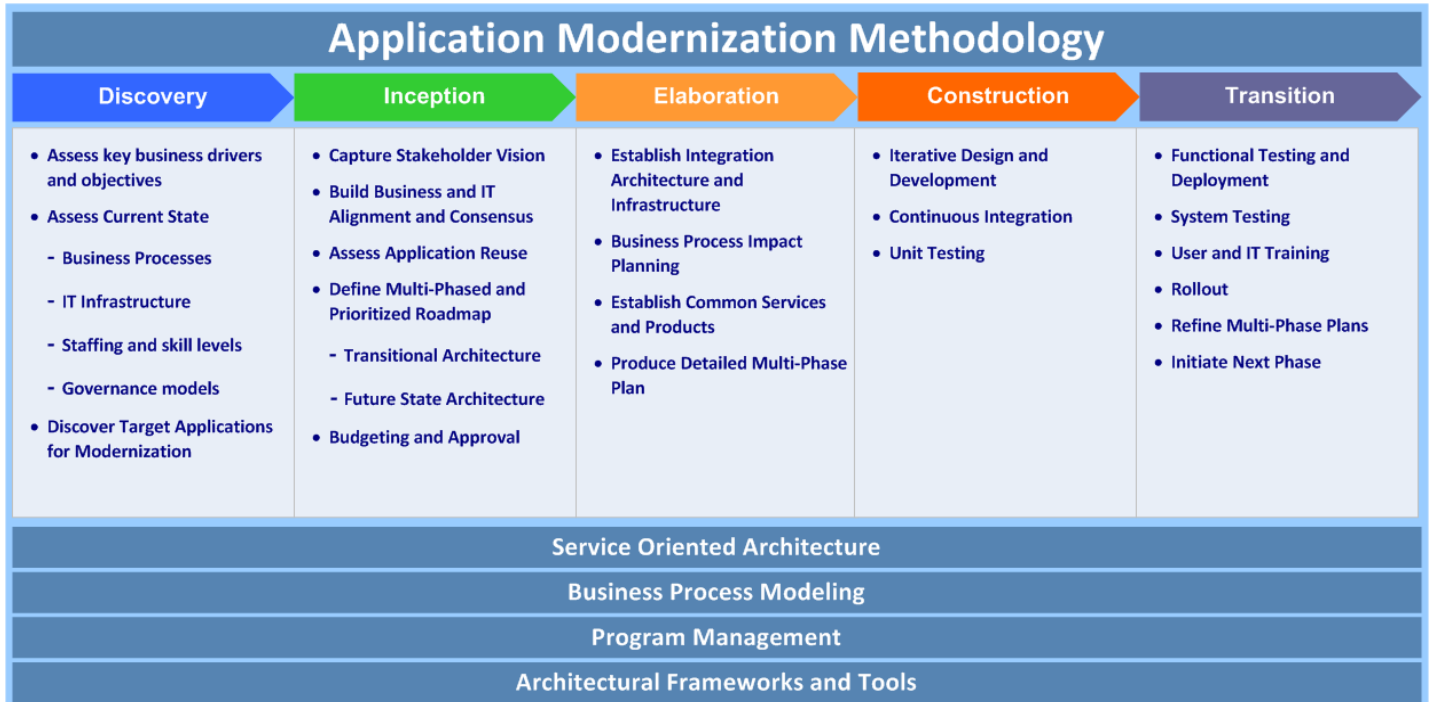


Figure 5

Application Modernization Technology

Today, there are more technology options available for executing application modernization implementations. Broadly speaking, application modernization technologies fall into several categories:

- Web enablement – Web portal servers provide easy ways to aggregate and integrate legacy applications into a single new, flexible user interface.
- SOA, BPM, integration and service enablement – Historically, custom integration development work has been very expensive, risky and under-designed. Today, there are a variety of tools and service-oriented approaches (such as Enterprise Service Bus (ESB) and Business Process Servers) that both support modern standards and expose legacy functions and data as web services. It is important to note that while SOA tools reduce costs and risks by focusing on standardized interfaces, their use still requires a unique mix of skills to understand current best practices and the technology and functionality of both the new and old worlds. In particular, performance and security aspects of your integration architecture require early and constant attention as functionality is exposed to other systems through integration points.
- Port/rewrite – More application development tools recognize the need to support a variety of environments and languages. For example, IBM has editions of its WebSphere Studio development products that integrate new Java J2EE development with mainframe and mid-range COBOL, CICS, RPG and other server programming, testing and deployment tools. There are also products that support mining, cataloging and reverse engineering of legacy application business logic and data.

Your modernization project may exploit one or more of these types of technologies depending on your roadmap. In addition, the modernized application will frequently be architected around the off-the-shelf capabilities of portals (e.g. content management or collaboration tools), business process management tools, or integration products (standardized B2B protocols).

Gartner identifies IBM's WebSphere family of products as the industry-leading set of development and integration products. A partial list of the WebSphere family of products is listed below.

Web Enablement

- IBM WebSphere Application Server
- IBM WebSphere Portal
- IBM WebSphere Host Access Transformation Services (HATS)

SOA Enablement and Integration

- IBM WebSphere MQ
- IBM WebSphere Portal
- IBM WebSphere Commerce
- IBM WebSphere Application Server
- IBM WebSphere DataPower SOA Appliances
- IBM WebSphere Virtual Enterprise
- IBM WebSphere Process Server
- IBM WebSphere Message Broker
- IBM WebSphere Enterprise Service Bus (ESB)
- IBM WebSphere Business Monitor
- IBM WebSphere Service Registry and Repository

Design and Development Tools

- IBM WebSphere Business Modeler
- Rational Application Developer
- IBM WebSphere Integration Developer
- IBM WebSphere Studio Enterprise Developer (WSED)
- IBM WebSphere Studio Asset Analyzer (WSAA)

Case Study: Federal Home Loan Bank

Summa recently helped the Federal Home Loan Bank of Pittsburgh (FHLBank) modernize its key online banking system. The existing architecture for this application was aging and no longer a strategic technology platform. Summa worked with FHLBank to move to an open-standards based architecture to integrate with other solutions and meet future needs. The modernization effort included architecture redesign, application development, system integration and implementation. To read the complete case study, visit www.summa-tech.com/pdf/FHLBankCaseStudy.pdf.

Summary

Application Modernization has become a common business practice for many companies who recognize that “throwing away” existing legacy applications and rewriting new ones from scratch is simply impractical. Developing a modernization strategy, outlined in a roadmap, provides organizations with a cost-effective method for improving the usability and realizing the best ROI for systems that contain core business processes. An effective roadmap can help companies make the most of their IT budget while maximizing the value to their business in a phased, cost effective manner that makes the best use of the capital available for IT projects. Vendor technology and industry standards allow companies to easily extend the reach and lifetime of legacy systems through web enablement and integration or to rapidly migrate to modern supported environments.

About The Author

Rick Kotermanski is the Chief Technology Officer for Summa. Rick serves as Summa’s technical lead for new practice and business development, in addition to his role as Principal Consultant leading client engagements. Rick has over 20 years of experience in the architecture, management, design and development of mission-critical application and product development projects.

About Summa

Since 1996, Summa has been providing high-impact IT consulting services and customized, commercial-grade software development for companies ranging from regional businesses to Global 2000 firms. Summa specializes in helping companies evaluate and implement IT modernization strategies to better meet their business objectives and is an industry-leading provider of Service Oriented Architecture (SOA), portal and BPM solutions. Summa is an IBM Premier Business Partner, a Microsoft Certified Partner and a member of the Oracle PartnerNetwork. For more information, visit www.summa-tech.com.