

A Summa Technologies, Inc. White Paper

# Strategy for Application Modernization



Summa Technologies, Inc.  
US Steel Tower, Suite 5320  
600 Grant Street  
Pittsburgh, PA 15219

412.258.3300  
[www.summa-tech.com](http://www.summa-tech.com)

The useful lifetime of a company's applications varies widely and is almost never what the original implementers had in mind. Custom or packaged applications frequently live long beyond their intended life – being morphed, twisted, pushed and pulled to hobble along for years, while others don't live to make it out of the gate.

The pace of business and technology changes are rarely aligned. Many applications were built with the best of intentions, but naiveté or lack of appropriate foresight into business and industry trends resulted in designs that are not agile enough to support changing business needs.

This paper discusses the primary factors that drive an organization's need to modernize applications. 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. The paper will review how to approach a modernization project by establishing and executing a realistic strategic roadmap that starts from the "as-is" 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 – factors such as staff inefficiencies, processing errors, lost customers, inaccurate data, broken business processes,

regulatory compliance warnings or high IT maintenance costs.

The following list outlines common reasons that organizations find themselves in need of application modernization.

- Unable to meet business objectives, due to factors such as missing functionality, lack of support or lengthy development cycles
- Struggling with regulatory compliance requirements from the SEC or with Sarbanes Oxley
- Inability to integrate with other internal or supplier/customer business systems
- 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

In addition to the previous list of drivers for modernization initiatives, some of the business benefits of modernizing include:

- Agility to quickly adapt to new business requirements
- Improved customer service
- Tighter integration with partners and/or suppliers
- Ability to exploit use of off-the-shelf products
- Reduced total cost of ownership
- Improved data quality
- Improved security management and control
- Improved operational efficiency

### Quotes From Managers Thinking About Application Modernization

“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?”

“Customers want to see their data. I want to ensure that they see ONLY their data. Our applications are complex. What choices do I have?”

### Getting Started: Establishing Roadmap Endpoints

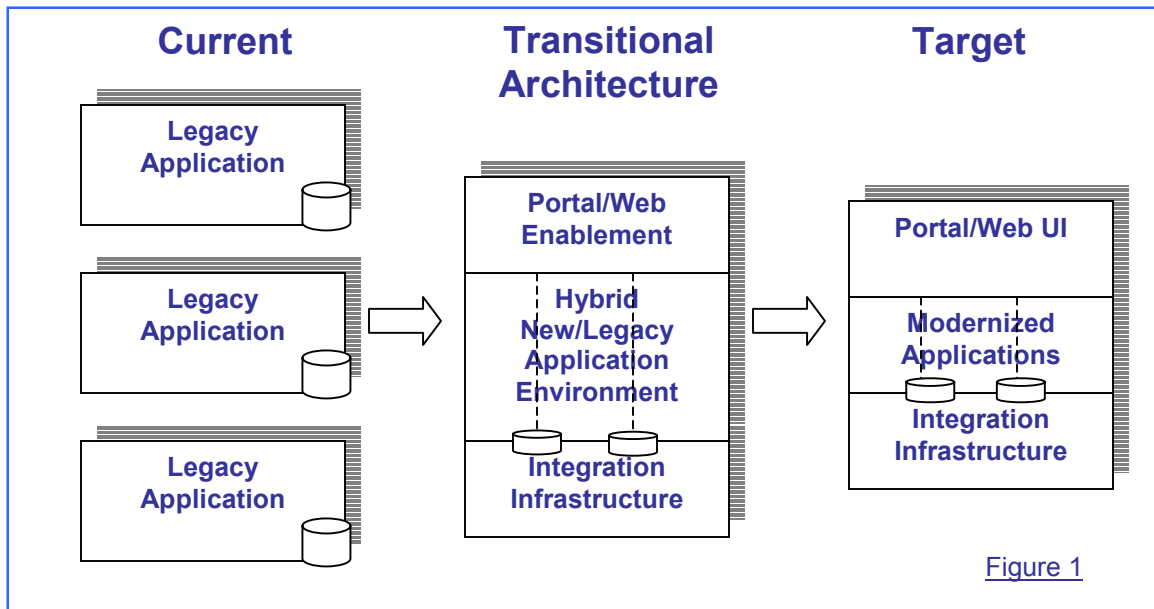
Modernizing a critical application typically presents big challenges that impact many parts of a company, so where do you start? The seemingly flip, but honest response is “it depends on where you are and where you want to go.”

To form a modernization roadmap, start with an assessment of the “as-is” state and the “to-be” vision. These two ends of the roadmap represent

more than a list of technologies. In addition to IT architectural considerations, it is important to consider the following factors as input to the roadmap:

- Directly associated and peripheral business goals, functionality and timelines
- Customer and end-user impact (positive and negative)
- Dependencies (on other applications, infrastructure or service providers)
- Integration points (internal and external to the company)
- Staffing impact – end users, IT and partners
- Capital availability and needs during the scope of the transition
- Organizational risk posture
- Known strengths and weaknesses of the current solution
- Any parallel IT and business initiatives that may factor into the modernization effort

Workshops, interviews and questionnaires are tools frequently 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.



### Modernization Approach

Figure 1 illustrates a roadmap view of the 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.

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

### The Target Reference Architecture

No matter which product vendors you are aligned with (either historic or future), to ensure an agile and scalable business solution, a 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 1. Establishing the integration strategy is a lot like city planning – analogous to building roads, water supplies, sewers and traffic systems – not glamorous or easy to see the value of the infrastructure, except when it’s not there or doesn’t work well.

Key goals of the integration approach include:

- Base the integration approach on standards-based solutions that support vendor interoperability and independence, broader availability of skills and tools, and a longer lifetime.
- 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, integratability, availability, extensibility and security are expensive or almost impossible to add in later.
- Establish a common IT asset governance model over integration practices – This ensures reusability and allows reuse of integration processes and best practices within and across applications. Myopia in the development of individual applications is one of the biggest impediments and hidden cost factors in modernization and integration efforts.

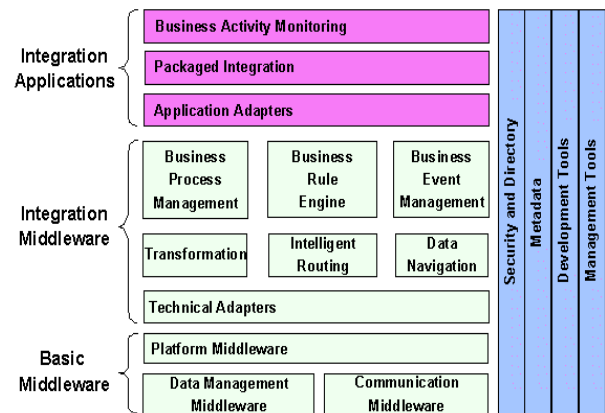
Some modernization projects only consist of laying a new web interface onto a legacy application or directly sharing data access with another application. While important, these types of projects require less up-front architecture and design.

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, it is recommended that you start with either an

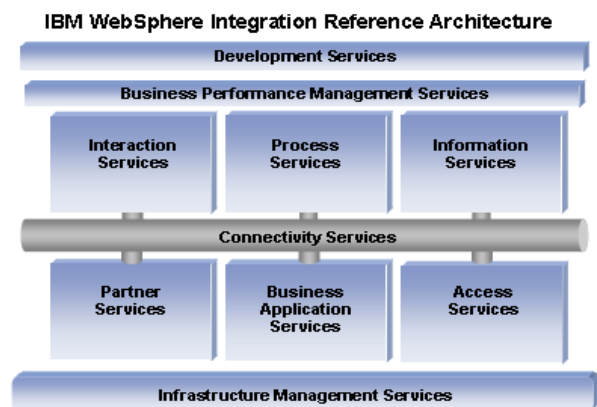
industry analyst or vendor defined reference model and design patterns. The reference model delegates responsibility of products and components within the IT landscape into slices of cooperating functionality based on the integration strategy.

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. Two example views of an IT reference architecture from Gartner and IBM are shown below in Figures 2 and 3.



Source: Gartner Research (March 2004)

Figure 2



Source: IBM (January 2005)

Figure 3

It is important to also 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.

### Establishing a Transitional Architecture

After establishing the endpoints, it is time to think about the road between point A and B. There are typically many options 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. If we are rebuilding city streets, should we upgrade the underground waterlines, sewer and telephone service while the hole is there? The answer is "it depends."

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

discussion of transition enabling products and technology approaches is included in the next section.)

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 or seasonal business activities may be a good or bad thing, depending on the purpose of the application and dependencies 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 contingency plans for various business and technical risks.

To accompany the "to-do" list, there is a list of transition planning and implementation practices to avoid:

- Short-term thinking – Consider long-term TCO (total cost of ownership) by ensuring that

maintainability, scalability and performance are addressed in the earliest project phases.

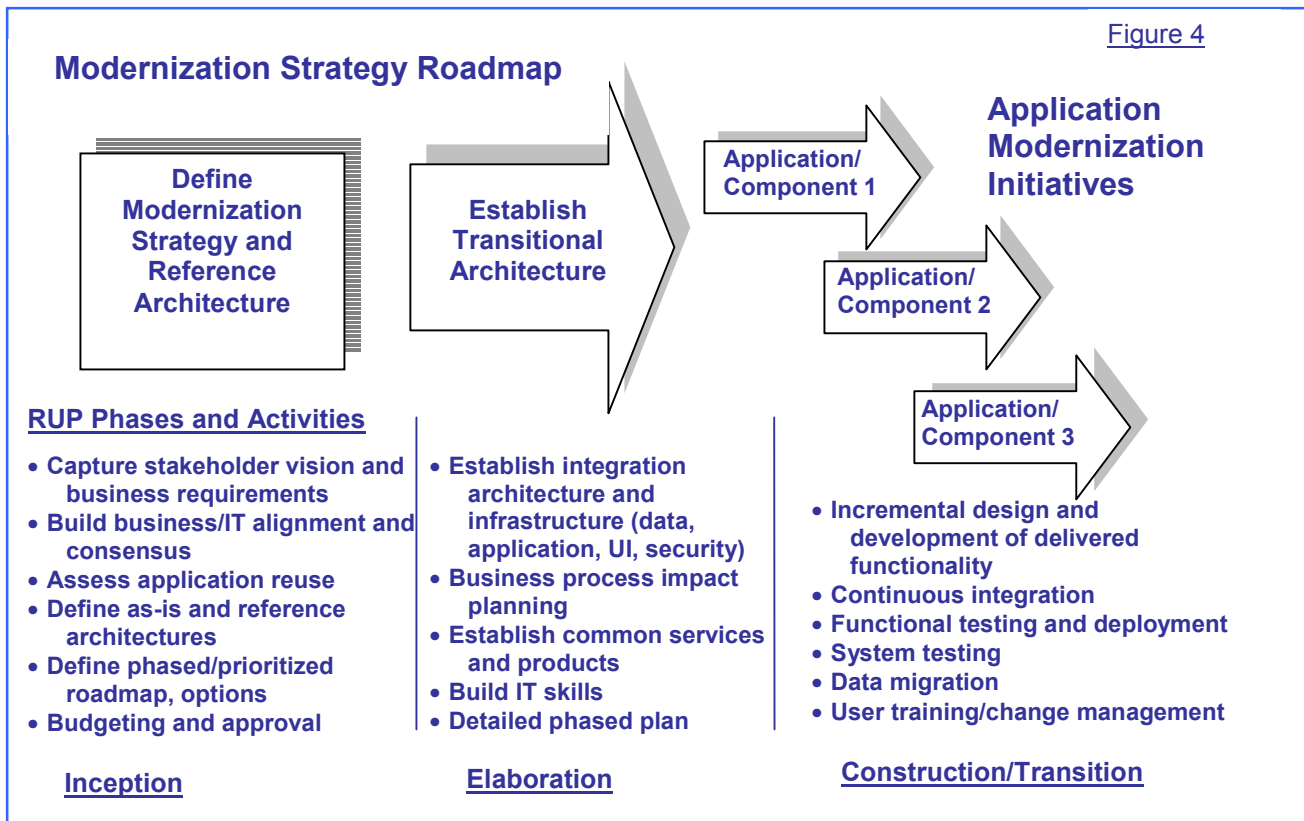
- Software migration tools that over-promise and under-deliver. Some automated code generation/migration tools greatly increase complexity and difficulty of long-term maintenance, over-dependence on vendor's proprietary products, dependency on difficult-to-find specialized skills and reliance on niche vendors. Frequently, simple translation will introduce subtle bugs and distance your staff from the application, making future changes very expensive.

Another aspect of the transition plan is deciding where to start from the technology perspective. Do you start at the "top" (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. (Further discussion of current product capabilities is outlined in a following section.)

### Execution of the Modernization Plan

The diagram in Figure 4 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. This is followed by a phased implementation of initiatives to modernizing, testing and deploying address applications and/or their components.



## 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. Advances in capabilities of web-based, screen-scraping products make development of these user interfaces easier than ever – sometimes with little to no development effort.
- Integration and service enablement – Historically, custom integration development work has been very expensive and risky (and typically under-estimated, ill-designed and poorly executed). Today, there are a variety of tools from various vendors that expose legacy functions and data as web services or through other connectivity products. It is important to note that while commercial integration 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 application 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, some or all of these types of technology depending on your roadmap. Also, the modernized application will frequently be architected around the off-the-shelf capabilities of portals (e.g. content management or collaboration tools), business process tools (workflow) or integration products (standardized B2B protocols). Gartner identifies IBM's WebSphere family of products as the industry-leading set of development and integration products, which are listed below.

### Web Enablement

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

### SOA Enablement and Integration

- IBM Enterprise Service Bus Offerings
  - WebSphere MQ
  - WebSphere Business Integration Message Broker
  - WebSphere Business Integration Server Foundation
  - WebSphere Business Integration Adaptors
  - WebSphere Application Server
- CICS Transaction Gateway / JCA or CICS SOAP Support
- IBM WebSphere HATS (Host Access Transformation Services)

### Design and Development Tools

- Rational Application Developer
- WebSphere Studio Developer Client for iSeries (WDSCi)
- WebSphere Studio Enterprise Developer (WSED)
- WebSphere Studio Asset Analyzer (WSAA)

## Summary

Application Modernization has become a standard 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 Principal Technologist for Summa Technologies, responsible for leadership and execution of Summa’s Technology Strategy and Practice Development. Rick has more than 20 years of experience in the architecture, management, design and implementation of mission-critical applications. His experience includes end-to-end responsibilities on application modernization, portal, business integration and product development projects using a variety of technologies.

## About Summa Technologies

Since 1996, Summa has been creating commercial-grade, custom IT applications for organizations of all sizes, including Fortune 100 companies. Our solutions provide web-accessible interfaces to mission-critical business applications, convert applications from one set of technologies to another, connect applications together or improve performance, reliability and quality. And all of our solutions work, the first time, because we take the time to understand the business problem before we recommend and implement the answer. Want to know more? Contact us at 412.258.3300 or visit [www.summa-tech.com](http://www.summa-tech.com).