CHAPTER 1

Introduction to the SOA Business Model

Service-oriented architecture (SOA) is a concept whose time has come. SOA is garnering great hype for such a simple concept, and we are here to tell you that SOA is more than hype. It is a concept with great promise for your information technology (IT) operations, for your business operations, and for your organization as a whole. We must remember, though, that SOA is a concept. Before we put our simple definition of SOA on the table, let's discuss what SOA is not.

SOA is not a product. SOA is not a solution. SOA is not a technology. SOA cannot be reduced to vendors' software products, much as they would like you to believe. SOA is not a quick fix for the IT complexity that has accumulated over 30-plus years. And finally, SOA does not address every IT challenge facing business and IT executives today. However, with proper planning and execution, SOA will deliver compelling business benefits to your organization in the short, medium, and long term. SOA is the right model for IT today, and for IT in the future. So, what is an SOA?

> *SOA is a conceptual business architecture where business functionality, or application logic, is made available to SOA users, or consumers, as shared, reusable services on an IT network. "Services" in an SOA are modules of business or application functionality with exposed interfaces, and are invoked by messages.*

ELEMENTS OF AN SOA

An SOA has many moving parts, not the least of which is the enabling technology that makes it work. The following list represents the essential ingredients of a successful SOA. Each ingredient is explained in the sections that follow.

- Conceptual SOA vision
- Services
- Enabling technology
- SOA governance and policies
- SOA metrics
- Organizational and behavioral model

Conceptual SOA Vision

An SOA is a business concept, an idea or approach, of how IT functionality can be planned, designed, and delivered as modular business services to achieve specific business benefits. The conceptual SOA vision includes clearly defined business, IT and architectural goals, and a governance model and policies to help enforce standards and technical requirements of the SOA over time. This is the definition of an SOA target state, the goal to be achieved over time.

Services

Yes, an SOA needs services, which as we said, means *all* possible services in the organization. Along with services comes a services design model to assure reusability, interoperability, and integration across all business processes and technology platforms. Services are the central artifact of an SOA. Services are the primary architectural asset of an SOA. As such, they merit significant attention throughout this book and throughout an organization's migration toward SOA through many projects and initiatives, each of which will most likely contribute services to the SOA over time.

Enabling Technology

While the technology of Web services and SOA generates lots of press, it is probably the easiest area to implement despite the vendor flux and standards volatility for various categories of technology solutions. The technology is essential to support realization of your SOA vision. However, the enabling technology is *not* your SOA. The enabling technology must be implemented to accomplish two objectives: (1) It must allow your services to operate reliably and securely in your enterprise in support of your stated business objectives; and (2) it must enable you to carry forward your existing IT architecture as well as enable your legacy systems to be leveraged to support your SOA goals. In many organizations, legacy mainframe systems and other applications are major contributors of services to an SOA.

SOA Governance and Policies

An SOA conceptual architecture cannot be realized unless it is communicated to the constituents of the SOA—business users, developers, architects, business and IT executives, and business analysts. In addition, communicating your SOA conceptual architecture to close trading partners is also advised. However, telling your SOA constituents what your conceptual architecture, vision, and goals are is one thing. Enforcing conformance to your SOA conceptual architecture, vision, and goals is another matter. SOA is not a big bang implementation model that we expect from large, packaged software applications. SOA is achieved incrementally through time at the project level by continuously defining and enforcing the standards that it will be based on. These standards are the policies that in the aggregate define your SOA conceptual architecture and, when implemented, help your organization achieve its SOA vision and business goals. An SOA governance model defines the various governance processes, organizational roles and responsibilities, standards and policies that must be adhered to in your SOA conceptual architecture.

Metrics

SOAs require a battery of metrics in order to measure the results you are achieving. These metrics include fine-grained metrics, such as service-level agreements (SLAs) for individual services, as well as usage metrics, policy conformance metrics, developer metrics, business and return on investment (ROI) metrics, and process metrics. Plan your metrics early, and don't forget them when you go live with services. You'll want the data, count on it.

Organizational and Behavioral Model

Your current IT architecture is the result of years of organizational behaviors, business decisions, and architectural choices. In order to achieve SOA, behavioral and organizational considerations must be understood and changed first; then over time will come gradual migration toward your SOA vision and goals. New organizational models and behavioral models will be essential to your SOA success.

SOA: BEHAVIOR AND CULTURE

SOAs contain a substantial amount of behavioral content because these initiatives are process-driven and span organizational boundaries. The "soft issues" of an SOA strategy must address the organizational issues and challenges that may help or inhibit SOA adoption, such as services ownership, the business and IT relationship, budgeting practices, and more. Organizational, cultural, and process issues thread through several facets of an SOA initiative. How do you organize your enterprise architecture functions and roles to support an SOA? How do you organize your developer resources to help ensure the realization of the goals and performance of your SOA initiative? What is the optimal IT structure for an SOA? Is centralized IT better? Or is a centralized enterprise architecture team optimal, supported by distributed developers embedded within specific business units? What are the skills, roles, and competencies of your architecture organization that will facilitate migration to and attainment of your SOA?

In addition, cultural and behavioral aspects are crucial to achieving SOA success. We will use a metaphor here. Imagine you're an

archaeologist. You're examining the artifacts of a long-since decimated culture—the physical remains, artifacts, tools, cooking utensils, and so forth—to ultimately make inferences about the behaviors that caused these artifacts to be frozen in their earthen matrix in the way that you've discovered them. That's what archaeologists do. They attempt to derive behavior from the physical remains and artifacts. Now, fast forward to your current IT architecture. It is comprised of legacy mainframes, distributed systems, desktop systems, software and documentation, user manuals, data models and schemas, which are all artifacts that resulted from the accumulated behaviors of your organization through time. These behaviors were a result of business and IT strategy and the various choices and decisions that caused your IT architecture to develop into its current state. So, if you plan to achieve SOA, you have to begin with behavioral, cultural, and other organizational factors that will lead to SOA success, and then architect your way toward SOA. You must enable and reinforce the behaviors that are more likely to result in the desired architectural outcome: SOA. If you start with enabling technology without changing behavior, years from now you'll end up with another layer of technology that an IT archaeologist will have to interpret.

Exhibit 1.1 depicts these elements of an SOA according to our model. As you can see, the SOA strategy drives the governance model and policies. Services are at the center of the model because they are the central asset and organizing principle of an SOA. They are the key asset of an SOA. The enabling technology surrounds the services, within the framework of the SOA governance model and policies. The SOA governance model also drives the metrics, the SOA architecture process, and finally the behavior and culture that must be addressed to ultimately realize the business and technology benefits of SOA.

Although these elements represent the essential ingredients of an SOA, there is much more to it. What most organizations will find is that they need new ways of managing various business and IT processes to meet the demands of an SOA initiative. This book represents a collection of models required to implement SOA. But why is SOA such an important concept now, and why is there so much interest in it these days? Simple. SOA offers too many business and IT benefits for business executives to ignore. Competitive advantage is at stake with SOA. First movers will have it; SOA laggards will not.

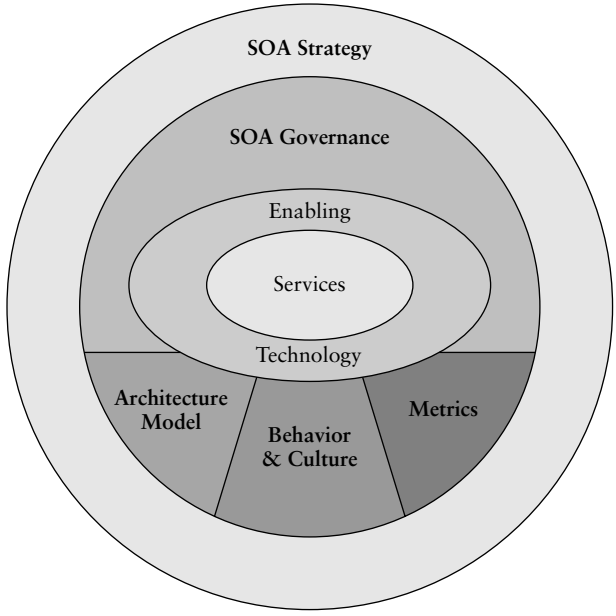

EXHIBIT 1.1 Elements of an SOA

NEW SOA CONCEPTUAL, ARCHITECTURAL, AND ORGANIZATIONAL MODELS

SOA initiatives will stress and in most cases break current operational and architectural models of IT organizations. SOA will require new ways of modeling and implementing various IT processes we have become accustomed to, such as services design models, integration models, reuse models, architecture processes, and enterprise architecture models. These other models, of course, augment the required SOA governance model.

SOA: ITS TIME HAS COME

One thing we know for sure, SOA is a concept whose time has come, and you do not need Jack Welch, Larry Ellison, Dr. Phil, or Oprah to tell you it's the right thing to do. If you are a business or IT executive and you are *not* thinking about how to implement an SOA in your

organization, you have already fallen behind your competitors. Your business and IT costs are higher. Your time to market is slower for new products and services. Your ability to implement IT solutions in support of business goals lags behind your competitors. And your legacy IT architecture is like a boat anchor embedded in the seafloor. You are at the mercy of the tides with no control of your destiny because you are beholden to your existing IT architecture. You are unable shed your legacy burden: the fixed costs, the outdated technology platforms, and the skills required to sustain it.

Over the years, your IT architecture has accumulated layer upon layer of complexity. When client-server architectures dominated the IT industry, client-server applications were layered over your mainframe platforms. When the Internet era rose to ascendance, Web-centric platforms were added on top of client-server solutions. And as these evolved into *n*-tiered architectures, the layers of IT complexity built up, building more modern complexity on top of legacy complexity.

Chances are you addressed the legacy systems problem with integration middleware, such as enterprise application integration (EAI) platforms or similar solutions. And this became yet another layer of complexity, or, as Brent Carlson, chief technology officer of LogicLibrary, calls it, "YALOT" (yet another layer of technology). These middleware or integration platforms were supposed to solve the legacy system integration problem and help simplify your IT architecture, but the reverse was true. These platforms became part of the very same problem, just more expensively and equally proprietarily.

Although IT architecture through the years is accretive, and nothing seems to ever go away, there are ways to "architect" your way out of this conundrum. One approach is to rip out all the legacy applications and replace them with modern ones. This rip-and-replace model is too expensive for most organizations. In addition, often it is not worth the risk and effort of replacing these still-working legacy systems with new software that will require significant modifications to suit your business model and business processes.

Another approach is to rewrite or refactor your legacy systems for modern application server platforms, such as J2EE or .NET. Even though rewriting systems is also very expensive, at least you know they will match your business processes when you are through. Like the rip-and-replace model, this approach usually is avoided, however,

not because it is not the right thing to do, but because it is expensive and difficult to cost justify to the business.

But there is a way out of this mess that avoids big bang system rewrites and expensive enterprise software projects. The way is *service-oriented architecture*. SOA is a simple concept, one that has the potential to alleviate many long-standing IT challenges and enable many coveted business goals that have until now been very elusive. SOA, by introducing a services layer into an existing IT architecture, can provide opportunities to isolate areas or elements of the architecture that are problematic, failure prone, or cost prohibitive. The services layering approach can enable the isolation, replacement, and/or potential consolidation of these architecture challenges while enabling the flexibility of reusable services. How often have business executives pronounced a desire to become more agile? When has time-to-market *not* been a mission-critical business requirement? Yet more often than not, these lofty business goals are constrained by outdated IT systems and incapable business processes that are subservient to tradition as well as to the digital concrete of today's enterprise software applications.

Why SOA Now?

SOA offers an avenue out of myriad business and IT challenges. However, before you leap into the SOA fray, you must understand a few things about it. First, SOA is not a new concept. It has, however, been refreshed with the advent of Web services, which have achieved more consensus from the vendor community than has been achieved in the history of computing.

SOA has also achieved trend status because of the degree of dissatisfaction that IT and business executives share with the current state of IT within their enterprises. Chief executives (CEOs) are fed up with hearing why they cannot expand into a new geographical location because the IT systems are not ready yet. They don't want to hear why the enterprise resource planning (ERP) system that cost $30 million will not support the new business process targeted to launch in six months. Chief financial officers (CFOs) are tired of waiting for regulatory compliance issues to be resolved, and they certainly are not pleased with an overbudget IT organization. Chief operating

officers (COOs) resent being told they cannot get a report because data are spread across three different systems, all on different computing platforms. Chief technology officers (CTOs) are fed up with vendors pushing more new technology when the old technology is still underutilized and operating as islands of functionality. They are tired of the endless need to keep integrating systems when the integration models themselves become part of the problem—more legacy silos to maintain. And chief information officers (CIOs)? They are tired of explaining the same problems time and time again. They're tired of having their budgets cut. They wish they had more funding for strategic projects instead of being hamstrung by having 80 to 90% of their budget committed to maintaining legacy systems. CIOs could do much more for the business if they could shed their legacy systems and focus on forward-looking strategic solutions for the business. There has to be a way out of this quandary. SOA could well be an answer. SOA is not new, but it's here to stay. SOA is finally achievable thanks to three major factors:

1. *Standards consensus.* Microsoft and IBM agreed, and the rest fell in line.
2. *SOA enabling technology.* Finally, implementing standards-based services is possible and affordable.
3. *Integration fatigue.* There has to be a better way to achieve application and business integration.

Standards Consensus

For the first time in the history of IT, there is widespread agreement on major SOA and Web services standards by all IT vendors. This nearly unanimous agreement means that whether you move now or later, you most certainly are going to be using services in your organization. Your software and platform vendors are going to take you there whether you want to go or not. Our advice? Preempt your vendors with your own SOA strategy and roadmap. Rapidly accumulate SOA experience. And be prepared to fend off any proprietary platform-specific approaches to services. Implement industry standards in your SOA governance model and in specific policies that will govern services identification, design, and implementation. You may

have to dedicate some internal resources to tracking relevant standards, but the ROI on standards will be well worth it.

SOA Tools and Infrastructure

With the advent of new tools and infrastructure solutions that enable SOA and services in a cross-platform, reusable, and interoperable fashion, SOA is real. This is perhaps the most significant departure from previous SOA implementations, such as CORBA (Common Object Request Broker Architecture), COM/DCOM (Common Object Model/Distributed Common Object Model), DCE (Distributed Computing Environment), and other proprietary schemes for reusable services. Interoperability for services is largely due to the standards for Web services, primarily SOAP for messaging and WSDL (Web services description language) for service descriptions. The variety of tools for legacy systems enablement, services development and exposure, Web services management, and multiple run-time environments for services have made the SOA industry very interesting to watch. There are as many ways to enable services and SOA as there are legacy systems and platforms in your architecture. Of course, bear in mind that we refer to the general case of "services" in this book. Some of your services will be Web services based on XML (eXtensible Markup Language), SOAP, and WSDL, as well as the extended WS-* standards. However, do not limit your SOA total value by examining and considering Web services only. Think services first, and then specific implementation models later.

Integration Fatigue: "There Has to Be a Better Way"

IT "business as usual" is over. Business and IT executives are frustrated with the lack of integration of their internal systems, with their business processes, with their trading partners, and with their customers. We call this *integration fatigue*. The business is demanding change, and IT executives know there has to be a better way. The frustration with IT as we know it is at an all-time high. IT budgets continue to be stressed, and they rarely increase. The majority of IT budgets are focused on maintaining current systems and keeping the

lights on; very little IT budget is focused on strategic initiatives that may pay future dividends. It is a do-more-with-less environment.

Business continues to change while IT is saddled with maintaining the systems and architectures of the past. IT doesn't have the luxury of eliminating its legacy or underperforming assets. Those very IT assets contain business logic that is most likely running a mission-critical portion of the organization's business. Yet while the business logic is mission critical, nonetheless the logic and data are locked up within individual silos of systems and technology. You cannot afford to rewrite the application, and your integration strategy has proved to be supremely costly to implement and maintain. There has to be a better way, and there is. It's called *service-oriented architecture*.

SOA: EVOLVED ENTERPRISE INTEGRATION

A major impetus for SOA initiatives is solving the age-old problem of integration. For many executives, SOA holds the potential to eliminate, via industry standards and modern tools, the proprietary integration model they've become accustomed to. According to many analyst estimates, up to 30% of a typical IT budget is allocated to integration activities. What would business be like if there was less integration, or, rather, if the integration that was performed was directly related to process integration, enterprise integration, and mergers and acquisitions (M&A) integration? In other words, value-added integration. How would spending less money on integration change an organization's competitive advantage? Could that budget be shifted to more strategic projects?

Origins of the Integration Problem

Where did all this IT complexity come from? Why is 80 to 90% of your IT budget focused backward on maintaining the past rather than looking ahead to supporting the future? This "rearview mirror budgeting" problem is legendary among CIOs and is partly responsible for the lack of strategic IT investment by CIOs today. How backward committed is your IT budget? What percent of the IT budget is allocated to maintaining your legacy investments rather than focused

on forward-facing initiatives that may move the organization ahead? This is a real challenge for both business and IT executives today. If you feel as if you're managing your IT budgets using a rearview mirror, you're not alone.

The demand for IT and process integration is driven by business requirements, such as:

- Increased M&A activity
- Corporate reorganization or restructuring
- Application and/or system consolidation
- Data integration and data warehousing initiatives
- New business strategies leveraging current systems for new processes
- Achieving regulatory compliance (e.g., Sarbanes-Oxley, or HIPAA, the Health Insurance Portability and Accountability Act of 1996)
- Streamlining of business processes to improve productivity

Addressing the business drivers for integration is a great impetus for SOA and Web services. At what point does IT complexity become an obstacle to business goals and an impediment to achieving IT's goals? We believe that complexity becomes intolerable when organizations are considering or taking these actions:

- Hiring a chief architect
- Creating a central architecture team
- Acquiring or developing your own enterprise application integration (EAI) software
- Creating an internal integration team, or a middleware organization, to help solve the integration challenges for your organization

Now, this does not mean that hiring a chief architect is a bad thing. More than likely it is a good thing. Chief architects can help address the architectural complexity and pain your organization is facing. The other actions just listed also can help address these areas. The problems are symptoms of the IT challenges that SOA could address. If you have considered or taken one or more of these actions, your organization is at the point where the integration burden is consuming IT resources, compounding the existing complexity problem, and inhibiting business and IT effectiveness. You most

likely have a rearview-mirror IT budget, and you are ready to try a new approach.

Stop Integrating; Service-Enable Instead

Stop integrating now. What we mean is stop integrating the way you have been using proprietary middleware solutions, homegrown point-to-point integration techniques, and tactical integration approaches that are doomed from the outset. These techniques are almost always going to break and require a significant ongoing maintenance burden from the organization.

An organization should integrate without integrating. Eliminate all point-to-point integration projects in your enterprise and rearchitect these initiatives from an SOA point of view. Inventory the integration solutions in your organization. Identify the IT budget allocated to these solutions and projects, including support staff, maintenance, and infrastructure. Determine how many of these integration efforts can be eliminated using reusable services in an SOA. (Integration projects are very good services opportunities. Chapters 3, 4, and 5 discuss the process of identifying and modeling services opportunities for an SOA initiative.) Identify the consumers of these integrations and determine their satisfaction with the current approach. How often do these integrations break? How often must they be enhanced or modified to support changing business needs? Do the n^2 math to determine how many interfaces can be eliminated with a services approach to integration. There will be a cost savings when comparing SOA and services to your current integration strategy. A services approach will be a more flexible, reusable approach than the point-to-point model you've been using. Although service-oriented integration will require more discipline and planning than previous integration paradigms, the results will be well worth the investment. Stop integrating now. Service-enable instead.

Exploring an SOA Business Scenario Exhibit 1.2 depicts a hypothetical approach to SOA as an alternative to conventional integration. This exhibit shows a fictional insurance company with three business units—Group, Voluntary, and Individual—at the top. Corresponding to the three business units are their own collections of systems and

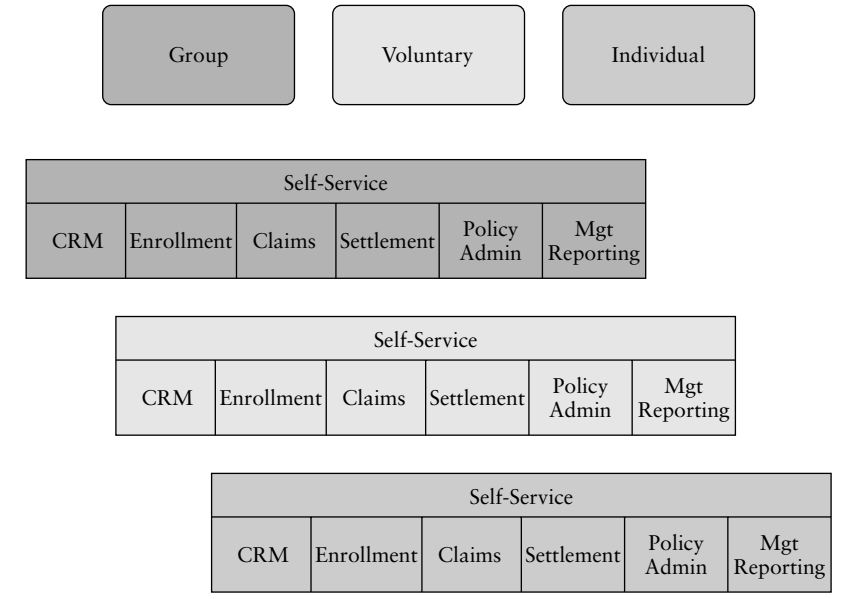

EXHIBIT 1.2 Typical IT System Silos

duplicate IT capabilities, accumulated over the years, and purportedly so unique that the business units must have their own systems to accomplish the very same business processes.

Each business unit has duplicate systems for sales and contact management, enrollment, claims, settlement, policy administration, and management reporting and controls. On top of these, all the business units have portal, intranet, and extranet capabilities to allow self-service for customers, agents, and brokers.

Identify SOA Business Opportunities Exhibit 1.3 examines this hypothetical business from an SOA perspective, exploring the potential business value that SOA may bring to this enterprise. What if the duplicate IT systems and business processes could be integrated and united as shared services by all three business units? What if the business processes across the three business units could be simplified to take advantage of shared IT services in an SOA model?

As shown in the exhibit, SOA offers several potential business benefits to this organization, such as product and process simplification,

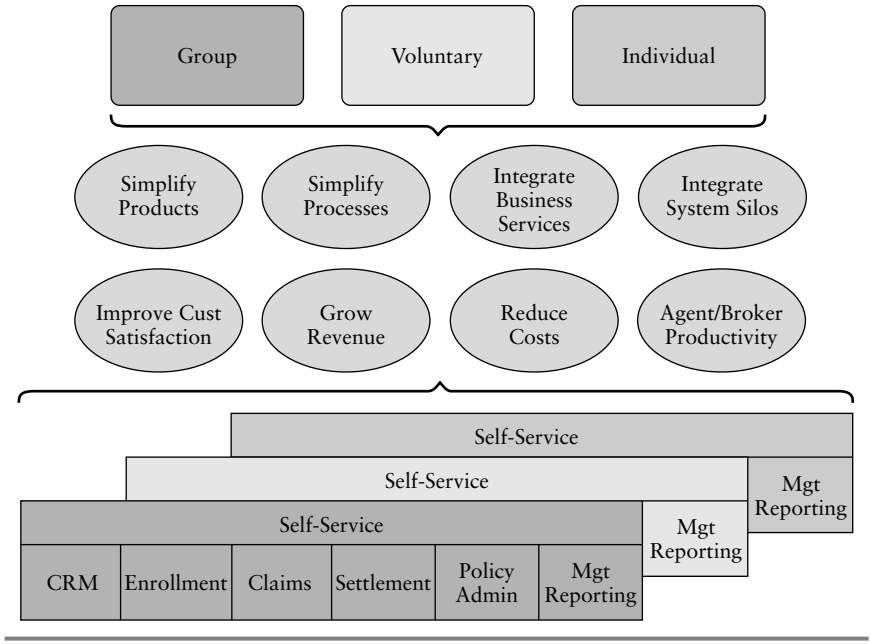

EXHIBIT 1.3 Identifying SOA Business Opportunities

integrated systems, integrated business services, better customer satisfaction, cost reductions, increased revenue and margin, and better agent and/or broker productivity. Of course, you cannot realize these potential benefits without performing proper analysis.

Examining the SOA Information Technology Potential Continuing the example, Exhibit 1.4 shows specific SOA opportunities that may apply to this insurance company. These opportunities include integration, process orchestration, better interoperability, services reuse, improved IT productivity, and achieving a real-time, event-driven enterprise.

Again, proper analysis and SOA planning of the business and SOA opportunities will determine the unique SOA value that will apply to any given organization.

Seeking Reuse Value from an SOA Initiative Let us assume that services reuse is a key SOA driver for this insurance company. Our SOA strategy, planning, and analysis will identify significant services reuse benefits in many areas of this company, across the multiple business units,

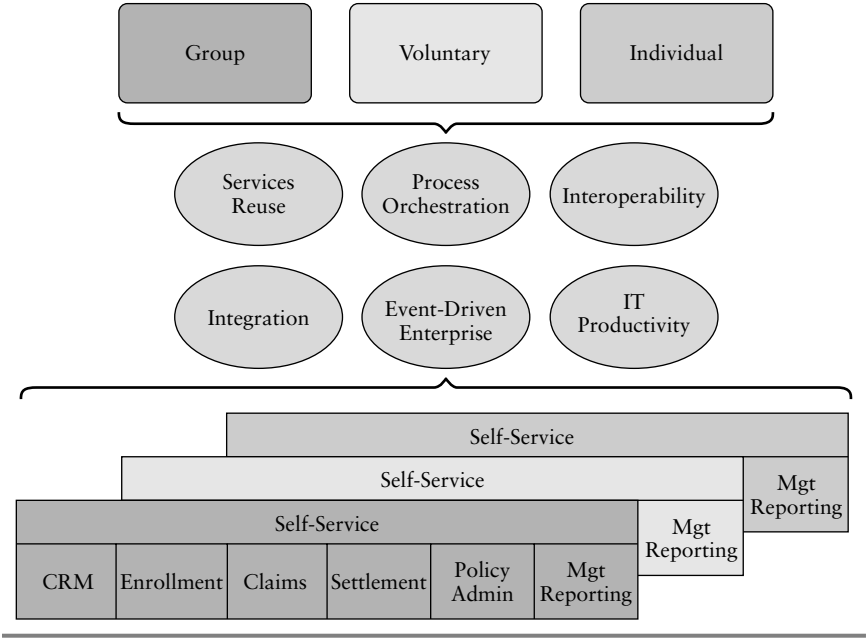

EXHIBIT 1.4 Identifying SOA Opportunities

and across the IT systems that support them. Exhibit 1.5 focuses this example on services reuse. In this insurance company, it was determined that reuse was essential to reducing costs, increasing IT productivity, and improving responsiveness to business demands.

This example shows how analysis might show that reuse of specific IT capabilities may offer tremendous business value. In this scenario, we focus on the enrollment process, which is supported by an excellent system but which can be improved by exposing its capabilities as shared reusable services in an SOA. Doing this allows reuse of a single IT service, enrollment, across three business units. This reuse eliminates IT complexity, increases IT productivity, and leads to simplified business processes in this organization. SOA applies as much to business processes and enabling better business functionality as it applies to IT systems—eliminating redundant systems, duplicate support infrastructure, and fragile and expensive integration strategies.

Service and Process Orchestration in an SOA Furthermore, this same organization is able to leverage its reusable services by orchestrating them

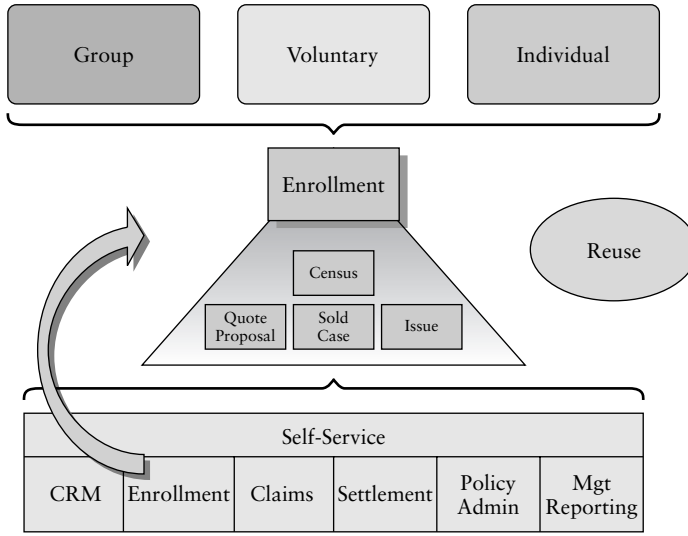

EXHIBIT 1.5 Services Reuse in an SOA

into business processes. Service and process orchestration extends reuse benefits by leveraging services as reusable composite applications and orchestrated process work flows. Exhibit 1.6 shows the enrollment process as a simplified series of services orchestrated into the business process work flow that the company decides is most reflective of how it wants to operate.

SOA allows an organization to stop integrating and instead recast its IT capabilities as shared reusable services. Once there are enough services to reuse in an SOA, further value can be harvested by orchestrating business processes based on these reusable services. In addition, the integration challenges that have historically plagued IT organizations can be avoided. Implementing the SOA scenario just described enables an organization to realize the many business benefits of SOA.

SOA: Competitive Advantage via Services

SOA is a concept that has direct business and competitive advantage implications for all organizations. For the business, SOA means increased customer satisfaction, real business agility, faster time to

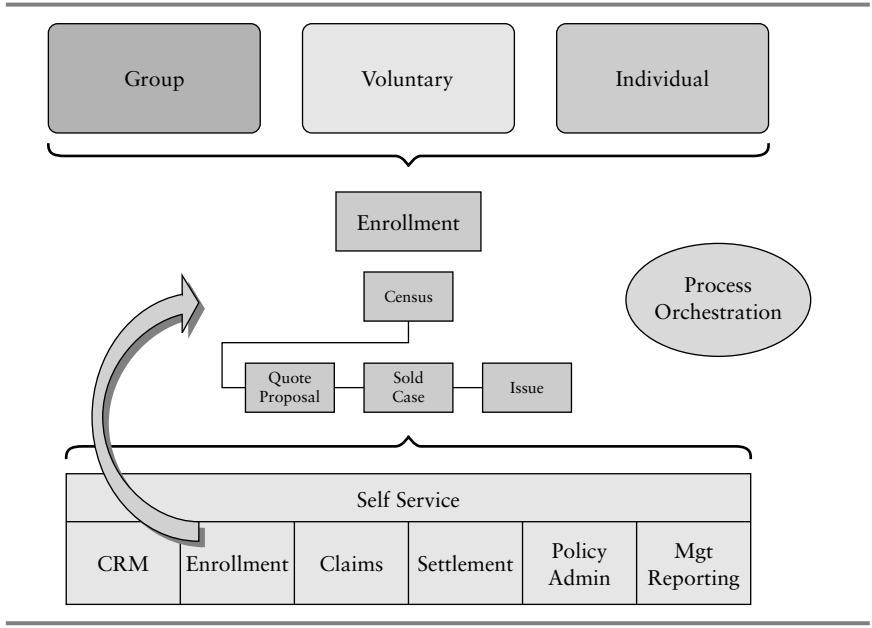

EXHIBIT 1.6 Orchestrating Processes in an SOA

market, ease of partnering, and lower business costs. Imagine that you can launch new products and services 30% faster than your competitors because you have eliminated friction within your enterprise, allowing better collaboration between your suppliers and your design engineers as well as better collaboration with your channel partners.

Your SOA has allowed you to speed up IT delivery to your business consumers. The time to implement needed system changes to support these new products has been cut by 25%, and you are able to make these changes using fewer resources: less development resources, less quality assurance resources, and less overall IT resources. Exhibit 1.7 depicts typical business benefits of SOA.

For IT organizations, SOA means greater productivity, faster time to market, greater asset reuse, agility, and lower IT costs. What if you could deliver IT services with less budget and few resources, while providing faster, higher-quality application support? SOA benefits an IT organization through faster application development, lower overall costs, greater software asset and services reuse, more

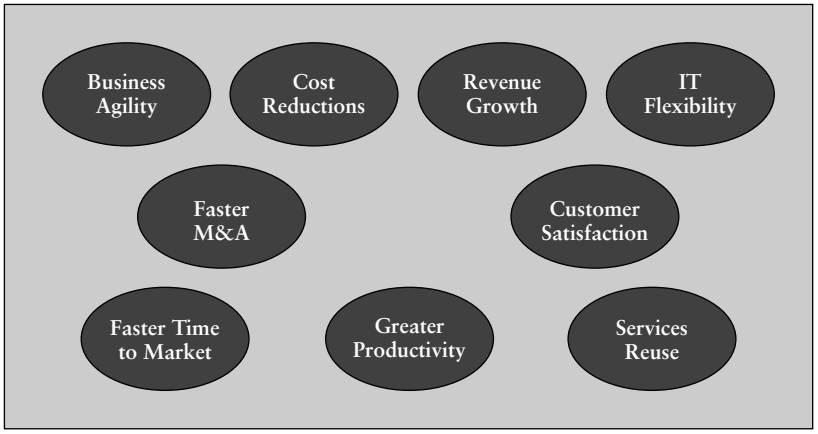

EXHIBIT 1.7 Business Benefits of SOA

repeatable software development processes, higher-quality applications via pretested components and validated Web services interfaces, and overall faster response to business customer requests for system enhancements and modifications.

SOA benefits the business through greater flexibility, faster time to market for business initiatives, faster response to business changes, and closer mapping of IT services to business needs. Many analysts see SOA as a mechanism to help finally achieve alignment of business goals and objectives with IT services and capabilities. Better alignment is partly attributable to the speed with which Web services can be developed and deployed in an SOA as well as the flexibility from leveraging proven, tested software components and Web services.

SOA also benefits an organization by abstracting business services (or Web services) from the specific technologies they were originally developed with and the platforms they were meant to run on. This twofold abstraction has two key benefits: (1) an organization can modify the technology architecture without mandating changes to the services available, and (2) the business community can change business processes without causing the ripple effect of changes to the services and underlying IT systems. Doing this leads to greater business agility and IT flexibility.

SOA MEANS "SERVICE-ORIENTED AGILITY"

SOA holds promise to finally make the word "agility" real for organizations. That's why the "A" in SOA, which stands for "architecture," could just as well stand for "agility." Service-oriented agility is one of the most oft-articulated goals of SOA. SOA enables business and IT agility along a number of dimensions. Although nearly every business and IT executive for the last 30 years has wistfully dreamed of achieving business agility, there has been little real progress toward that end save for a few exceptional firms. For most organizations, business agility is a vision without reality. Until now. SOA and services provide a means to achieving true business agility. Business agility can come from two broad forms: the ability to change business processes to meet changing market demands and customer requirements, and reduce costs; or the ability to execute business processes faster or launch new processes, products, and services faster. Agility and speed are both real and tangible benefits of migrating to SOA and reusable services.

SOA can help an organization unshackle its business processes and data from the IT systems that support or, in many cases, constrain them. Using a services approach and SOA, enterprise software systems will be decoupled from business processes through the use of business services. Business services will be defined as abstracted entities separate from the business logic that is locked within enterprise applications such as SAP, Oracle, Siebel, and other monolithic enterprise applications. When application logic is exposed as a business service, it becomes a shareable and reusable software asset, or *service*, that can be coupled with services from other applications to create new sources of business process value, completely new business processes, and even more efficient versions of existing processes using business process management (BPM) tools.

Service-Oriented Agility: Speed

One aspect of agility is speed. The ability of an organization to hasten its response to market changes or competitive threats, or to quickly preempt competitive moves from the competition, is clearly an advantage. Speed consists of two dimensions: the total elapsed

time of a business action or response, and the speed of the IT component of the business action or response. Enhancing speed could require installing a new system, developing a new system, running a new report, or whatever the specific business requirement is to support of the business.

If the software development cycle of an organization is too slow for the business to respond to market changes or competitive threats, then the business does not have agility, and clearly IT doesn't have it either. SOA can create agility through speed via faster application development, which in turn will contribute to speedier business responsiveness to market conditions, competitive threats, and customer requests.

Service-Oriented Agility: Flexibility and Range of Response

Service-oriented agility can also be expressed as flexibility, or by allowing a greater range of options for a competitive response and making that response easier. An IT architecture can by its very nature limit the range of options an organization has to respond to market opportunities and customer requests. However, an SOA may offer a greater range of options by reducing the fundamental unit of IT to a business service.

Business agility and IT flexibility are always mentioned in corporate documents, in annual reports, and by business executives when they talk to analysts and customers. Agility and flexibility are among the most discussed and yet least achieved goals in corporate history. Part of the problem comes from a failure to operationalize the terms so they can be implemented and to put metrics in place to help realize them.

SOA: Agility Focal Point

SOA represents an opportunity to regain the agility and flexibility that many organizations lost in the 1990s with enterprise software applications and point-to-point home-grown and commercial integration models. SOA allows the creation of a services layer that

resides between the business architecture of an organization and the IT architecture of an organization. We call this layer the agility focal point. This services layer is the decoupling abstraction layer that insulates business processes from IT changes and allows IT to change technology without changing business processes. Exhibit 1.8 depicts the concept of the agility focal point via an SOA.

An SOA implements the agility focal point concept by facilitating the flexing of the business and IT architectures—the SOA in particular—in response to business changes. SOA enables this strategic business capability, which allows an organization to compete based on agility, or *service-oriented agility*.

Competing on Service-Oriented Agility

With more IT flexibility and business agility, all organizations should be seeking faster time to market for new products and services; faster responses to business, competitive, and environmental change; and an overall better ability to quickly adapt both business processes and IT systems in support of change. In their seminal book *Competing Against Time*, Stalk and Hout discuss the notion of time-based competition and how those who are faster to market are more profitable than those who are not.[1] SOA enables many of these time-based

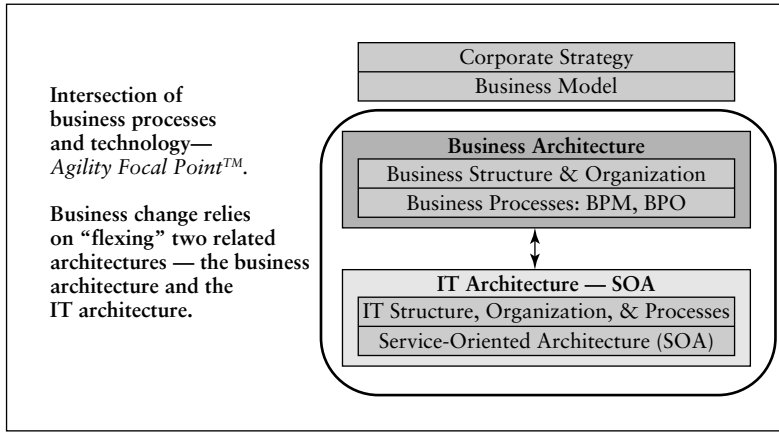

EXHIBIT 1.8 SOA and the Agility Focal Point

capabilities by eliminating many of the traditional IT and business barriers to change (e.g., inflexible business processes, business processes locked up within rigid IT systems, inflexible IT architectures with rearview-mirror commitments to legacy systems, etc.). SOA and services provide a business solution to the problem of adapting to change, and this business solution is based on both the business and IT being able to adapt and respond quickly.

Regaining IT Flexibility: Breaking the "Rearview Mirror" Paradox

Much as agility is the Holy Grail for business executives, flexibility is equally sought after and is equally elusive for IT executives. For IT executives, their continual challenge is supporting the years of accumulated legacy systems and infrastructure in the face of shrinking budgets. SOA and Services provide a pathway toward breaking free of the rearview-mirror budgeting paradox. First, in order to inject flexibility into your IT architecture, you do not have to rewrite or refactor every legacy application or enterprise application in your portfolio. You merely have to begin exposing portions of business functionality as services that match to business process requirements of the organization. Second, most of the services you will target initially in your SOA initiatives are contained within existing applications. The challenge is to expose the business functions as reusable services that can in time be combined with other services into composite applications, orchestrated processes, and BPM solutions. Third, SOA is an incremental architecture, meaning that it is not implemented or attained in a big bang model. SOA is achieved over time by defining and adhering to a body of architectural goals, standards, and design guidelines so that all services will interoperate over time within and, when necessary, outside of your enterprise. SOA is the "anti-enterprise application" in that it encourages freeing services from inflexible application architectures imposed by others, namely software vendors, and begins to define your vision of services and business processes to better match the way you want to conduct business. Finally, the concept of business services can allow the IT organization to insulate itself from the constraints imposed by both its legacy systems and its more modern applications. SOAs

future-proof your IT architecture. SOAs are built to accommodate change.

Why You *Must* Begin SOA Initiatives Now

With the analyst and media buzz about SOA, you may be asking yourself why you should believe it. What's so special about SOA that you need to invest your time and resources in this concept at this time? Why now, when you survived the past technology paradigms that were nascent attempts at SOA, namely CORBA, COM/DCOM, and others?

This time, the stars and moon are aligned in SOA's favor. We've covered many of these already, such as the unanimous agreement on the core standards of SOA, the cross-platform capabilities of SOA using Web services and these core standards, and the fact that IT maturity is now more able to assimilate the concept of SOA to drive a business result. There is so much discomfort with the current state of IT that something has to give, and in this case, it's the entire process of IT architecture and services delivery that has to be reconstructed.

SOA presents an opportunity to change the rules of the game. SOA will allow firms to compete using their SOA efforts along a number of business and IT dimensions. These firms will be applying SOA to their businesses to create service-oriented business.

These advantages will characterize SOA first movers:

- *Competitive advantage.* If you beat your competitors to the SOA punch, you will have achieved competitive advantage on a number of fronts, including business speed and agility, IT cost and delivery, and customer satisfaction.
- *SOA cycles of learning.* SOA first movers will gain the experience required to fend off IT vendors partners; you may as well preempt your vendors by getting in front of the SOA wave. If you're going to end up with services anyway, you may as well be prepared and ramp up your ability to consume and provide services.
- *Break the rearview-mirror budget crisis.* There are two ways out of this situation: (1) fix your architecture process to suit an agile, changing services world and (2) stop integrating and service-enable instead.

SOA first movers will be in a better position to compete in a world of services, where vendors, customers, and business partners will all eventually transact via SOA and services. This is a world of service-oriented business, or business-oriented architecture.

SERVICE-ORIENTED BUSINESS: SOA = BUSINESS-ORIENTED ARCHITECTURE

Although many organizations are seriously considering SOA, some are doing it from a pure IT perspective while others are really looking at a new way of running their businesses. They are pursuing the concept of service-oriented business, or the idea of running all aspects of their organization from a "services" perspective. This has broad ramifications for an organization.

IT is a major budget item in many firms, especially for financial services organizations. IT affects all aspects of these organizations in profound ways. IT affects the cost of everyday business transactions, which over time equates to billions in efficiencies. IT allows interdisciplinary collaboration across a business enterprise, which means better cooperation, better sharing of information, and a more united capability to compete in your markets. IT improves many diverse business processes and increases productivity across the organization. IT amplifies the productivity and efficiency of all business processes, both internally as well as those that are exposed to trading partners and customers.

Business Impact of SOA

The potential business impact of SOA is significant to an organization, but only if the appropriate business modeling and business context are considered during the planning and implementation of SOA. The range of business benefits is broad, and includes both specific and tangible benefits as well as less tangible yet more compelling benefits. Of these, one of the most interesting concepts is the notion of the *SOA Network Effect*. This concept was first developed in a *ComputerWorld* column to describe the importance of the soft issues relating to achieving service-oriented architectures, such as cultural

issues, behavioral issues, and organizational dynamics.[2] Are these as important as the technology of SOA? The answer from the field is that these soft issues are more important than the technology, and thus they are not all that "soft" as we would suppose. These soft issues are among the most difficult aspects of achieving SOA. Yet they are the key to an SOA's ultimate success or demise.

Characteristics of a Service-Oriented Business

A *service-oriented business* is an organization that has progressed with its SOA efforts such that its business really does operate using an SOA. SOA as an IT architectural strategy actually uses services as the "operating system" for the business and its business processes.

What if your CEO could acquire another firm and integrate its information and operations into the existing business and IT architecture without integration challenges? What if the acquired firm's business processes could integrate seamlessly with yours with zero latency? What if there was no integration effort required at all? What if the IT systems were "preintegrated," meaning that they could exchange information with one another without any incremental integration expenses and effort? What if business processes could be more quickly combined into new composite processes that represent the combined business entity better than those of the two individual business processes?

Imagine how business would be without two factors that have become ingrained in our expectations of IT today: the IT integration hurdles that attend every business initiative and the inevitable time lag between the need for the business initiative and the ability of the IT organization to deliver it on time and, more important, with zero time-to-market latency. This is the zero integration business enterprise.

How would this enhance the business case for a mergers-and-acquisition strategy? How would this improve the return on investment equation for any IT or business initiative? What is the value of faster time to market for any information-based initiative? That's the business case for a service-oriented business.

Exhibit 1.9 is a vision of how SOA becomes the foundation for a service-oriented business. It depicts how SOA can enable significant

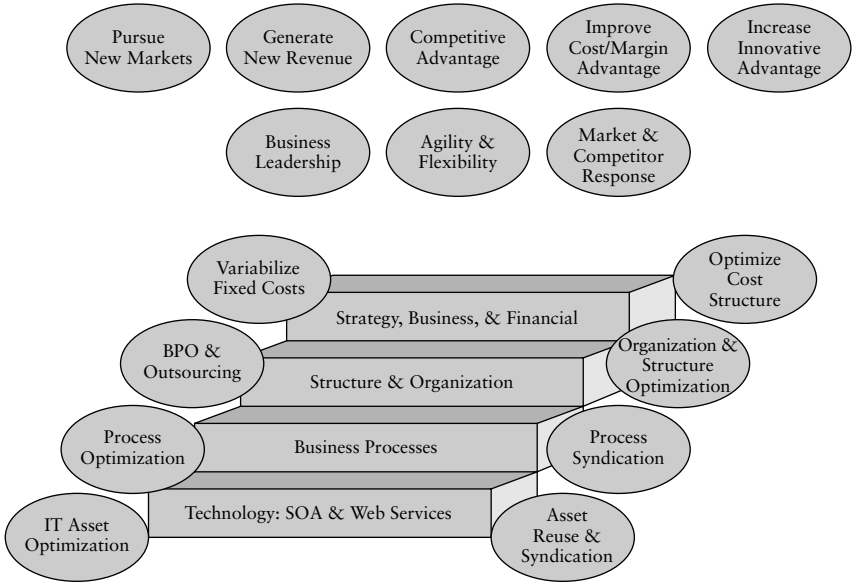

EXHIBIT 1.9 Service-Oriented Business

strategic business benefits through many tangible contributions. These include IT benefits, such as reuse and asset optimization that are often initial target benefits of most SOA efforts. However, as we progress from the IT level to business processes, organization and structure, and ultimately the strategic level where business and financial goals are critical, SOA has value to offer there as well.

CHARACTERISTICS OF A ZERO-INTEGRATION ENTERPRISE

Implementing SOA will allow tremendous reduction in integration expense and maintenance, such that we go so far as to call it the "zero-integration enterprise." A zero-integration enterprise is an organization whose business and IT teams are committed to the precepts of SOA. These organizations see the business value and strategic advantages of SOA, and are migrating their organization, processes, applications, and skills to support the concept of services,

specifically Web services. This organization has eliminated traditional integration models in favor of SOA and standards-based integration. Some characteristics of a zero-integration enterprise follow.

- This organization can launch new business initiatives faster than its competitors because less IT integration is required to support various business projects. This time-to-market benefit allows faster response to business conditions, customer requirements, competitive threats, and increased innovation.
- This organization has a higher return on assets and greater IT productivity than its peer companies due to the asset-related benefits of SOA, such as software component reuse, services reuse, and extending the capabilities of existing IT systems and infrastructure.
- This organization launches new software applications 35% faster than before with reduced quality assurance and testing effort due to component and services reuse. Building on proven software and services capabilities, the organization spends less time developing new code and more time focusing on business process issues.
- This organization uses the 30% of its IT budget previously used for integration projects to solve strategic business problems, such as improving customer satisfaction, reducing time to market for new products, and increasing sales through various IT initiatives.
- This organization implements concepts of agility and flexibility through its SOA initiatives. Agility is reality, and is measured by clear unambiguous metrics.
  - This organization has an agile business model that can quickly respond to business challenges, competitive threats, and customer needs.
  - This organization has greater customer focus deriving from reduced effort spent on internal integration and more effort spent on customer satisfaction, partner communication, and efficient business processes.
  - This organization has a business-focused IT organization that no longer must concern itself with assuring interoperability issues but rather can focus on forward-looking strategic issues.

- This organization has a flexible IT architecture, based on SOA and services, that facilitates superior business performance, enables world-class business processes, and is highly efficient with all corporate resources.
- This organization integrates without integrating, both internally and externally with customers and partners. This organization does not integrate, it service-enables instead.

WHAT ARE THE CHALLENGES OF SOA?

Now, with all of these benefits, where's the catch? Simple. SOA is difficult to implement, manage, and control. Not because of the technology, mind you, but due to the organizational, cultural, and behavioral aspects of SOA that contribute to success. And technically, although there has been great progress with regard to the standards, supporting tools, and development and run-time platforms, there are still issues to be resolved. These issues include support for long-running transactions, security concerns, and many others. However, the organizational, cultural, and governance issues far outweigh the technical aspects of implementing SOA. That's not a reason to avoid trying to achieve SOA, but it certainly is a reason to pay attention to many of the softer aspects of technology initiatives to ensure you can reap the rewards. A discussion of major SOA challenges organizations will face as they migrate to SOA follows.

Enterprise Architecture Model May Need Tuning for SOA

As many organizations consider their SOA approach, they will realize that the organization, processes, and disciplines of their enterprise architecture organization may require tuning to suit the requirements of SOA. Often the process of enterprise architecture is somewhat flawed, which helps explain the current state of IT architectures today: rigid IT architectures characterized by heavy carryover legacy systems, inflexible "digital concrete" of enterprise applications, and a portfolio of applications that demand integration software to make them work together.

Perhaps architects have been too focused on building things and getting them to work as opposed to building things that are flexible, reusable, and support the business: the things that are now most important for businesses today as they grapple with change and global forces. These are the new requirements of SOA.

SOA will fail unless the process of architecture is changed from one of static advice, creation of presentations, application blueprints, and architecture road maps to one of actively shaping and implementing flexible and reusable IT assets that support business processes. In other words, SOAs. A model for tuning the enterprise architecture process is presented in Chapter 8.

SOA Is Spatially and Temporally Distributed

One of the challenges of SOA is that it is not implemented all at once. Rather, it is achieved through many discrete projects across both space and time. This temporal and spatial distribution of SOA projects makes governance all the more critical to SOA success. SOA governance and enforceable policies are the keys to managing conformance to the SOA across geographic and time horizons.

SOA Is Organizationally Complex and Behaviorally Challenging

SOA is a complex goal to achieve. It is organizationally, behaviorally, and culturally challenging for most organizations. We describe an SOA behavioral model in Chapter 7 to help you anticipate and create the behavioral pattern your SOA will require.

SOA Requires Governance to Achieve and Manage

SOA requires a robust SOA governance model, clear and enforceable policies, and a way to implement SOA governance across all the lifecycle processes of an organization: enterprise architecture, services design, publishing, discovery, and run-time. SOA governance is essential to realize the ultimate business value of SOA.

SOA Is All About Services

Implementing SOA requires new approaches to identifying, modeling, and implementing reusable, interoperable services. Repeatable processes for determining appropriate granularity, version management, and enforcing design-time policies for services are essential. In fact, even the definition of services is important for many organizations.

The fundamental architectural unit of an SOA is a "service." Services, per our definition of SOA, are units of business capabilities, processes, or functions that are delivered in a repeatable way to consumers of those services. Consumers of services in an SOA can be developers, architects, and analysts, or they can be external customers, business partners, and internal business customers.

Without services, there is no SOA, and an SOA with services is useless unless there is actual consumption of the services that are available. Therefore, an organization can deliver or realize no SOA value unless the services in an SOA have real consumers using them for business reasons.

Services Identification, Modeling, and Design Challenges

Services are critical to an SOA. However, the process of identifying the right services for your organization is challenging. And what makes these services the "right" ones? This discussion is further complicated by questions about granularity, reuse, and other related services modeling and design issues. Chapters 3, 4, and 5 address these concerns using some innovative services modeling concepts. *These chapters will help you achieve the "SOA Three Rights": Identify the "right" services; build those services the "right" way; and run them on the "right" enabling technology stack.*

Where Are We Headed?

SOA is an iterative business approach. There is no single correct path to achieving SOA. Instead, there are multiple routes to the SOA goal. It is important to recognize the fundamentally iterative approach that

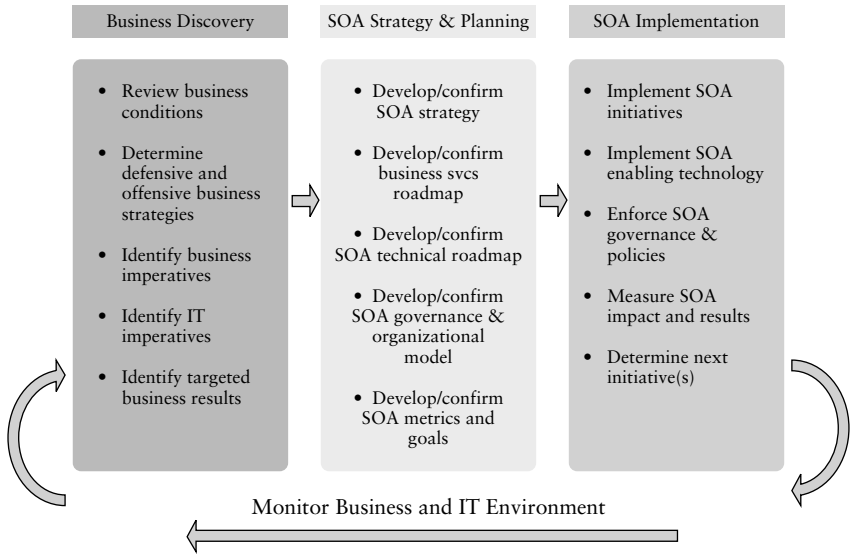

EXHIBIT 1.10 SOA Business Iteration Model
Source: AgilePath Corporation, copyright © 2005. Used with permission.

SOA requires to achieve its stated goals objectives and business results over time. This approach, which we call an SOA business iteration model, is depicted in Exhibit 1.10.

This model builds explicit business context into the SOA strategy and planning process, recognizing that SOA must be aligned to business and IT objectives as well as to the current urgencies of the organization at that particular moment in time.

Remember: SOA is a lifestyle change. It is a long-term commitment to achieving specific business objectives. That is why we wrote this book, and that presumably is why you are reading it: to learn how to plan, design, and implement SOA via reusable services to achieve clear business results.

NOTES

1. George Stalk Jr. and Thomas M. Hout, *Competing Against Time* (New York: The Free Press, 1990).
2. Eric A. Marks, "The SOA Network Effect: Technical and Cultural Issues Drive Value," ComputerWorld Online, August 16, 2004.