zapthink
white paper

JUSTIFYING & FUNDING YOUR SOA PROJECT

SOA BENEFITS & THE ROLE OF GOVERNANCE

ZapThink, LLC • 108 Woodlawn Road • Baltimore, MD 21210 • info@zapthink.com • www.zapthink.com

JUSTIFYING & FUNDING YOUR SOA PROJECT

SOA BENEFITS & THE ROLE OF GOVERNANCE

September 2007

Analyst: Jason Bloomberg

Abstract

Many organizations struggle to build the business case for implementing Service-Oriented Architecture (SOA)—not because SOA doesn't provide numerous benefits to the organization, but rather because they don't properly identify the business problems in their organization that SOA would be particularly well suited to address.

This paper addresses this deficiency by delineating the most important business benefits of SOA: reduction in the cost of integration, achieving asset reuse, increasing business visibility, and achieving business agility. Implementing SOA to achieve these benefits, however, requires many capabilities that fall under the broad umbrella of SOA governance, including visibility into IT assets, change management, enforcement of best practices, measurement of effectiveness, collaboration capabilities, lifecycle management, and open standards support.

I. The Business Benefits of SOA

The first question many architects ask as they embark on their Service-Oriented Architecture (SOA) roadmap is, "I can see that SOA is the architecture of the future. How do I sell it to the business?" The problem is, that's the wrong question. Any architect who begins with trying to convince line of business executives that SOA is worth investing in will invariably meet resistance or even scorn. The right question should follow the following lines: "these are the business problems; how do I solve them?"

In order to have a solution, you must first begin with the business problem. If all your problems were nails, then a hammer would be sufficient.

There are two critically important lessons for architects to learn here. First, in order to have a solution, you must first begin with the business problem. If the business doesn't have a problem that's causing them pain, or you don't understand what the problem is, then they won't invest in a solution for it. Secondly, the solutions that the information technology (IT) department brings to bear to solve the business's problems depend on the particular problem. To add a new twist to the old saying, if all your problems were nails, then a hammer would be sufficient. But in the business world, there are all manner of different types of issues that cause the business pain, and as a result, it's important for architects to have many tools in their toolbelts.

SOA, then, provides some great tools for the architects' toolbelts—but it's not just one tool, and it won't give you all the tools you need to solve every problem, either. As a result, for architects to be successful with SOA, they must understand what problems SOA is good at solving, separate those problems from others that they should best solve some other way, and finally, understand which solutions go to which business issues. Only then can architects successfully build the business case for SOA in their organizations.

II. Reducing Integration Expense

The first business benefit that SOA provides organizations is the most straightforward—reduction in costs, namely of integration. Reducing expense is often the first business benefit organizations target for their SOA initiatives, because it offers the most straightforward, short-term return on investment (ROI). In fact, in situations where there is skepticism on the part of business management, building a SOA value proposition based upon quantifying the cost savings that SOA can provide can build a sufficient case to initiate the SOA initiative. Once the organization successfully executes on those cost savings, then, it becomes much simpler to take the next step with the SOA initiative and achieve other benefits as well.

Leveraging Web Services

Even though implementing Web Services is neither necessary nor sufficient for implementing SOA, leveraging open standards to reduce integration expense is an important precursor to SOA. It's important to follow a proven methodology for constructing and implementing Web Services, both to achieve this short-term benefit as well as to properly prepare for SOA. Organizations who do not follow certain SOA best practices when implementing Web Services often end up with redundant, incompatible, or rogue Web Services. For those organizations, moving to SOA will necessitate reworking existing Web Services, which reduces or eliminates the cost saving benefit.

Web Services promise a level of compatibility across multiple technology platforms. Web Services offer a set of industry- standard protocols for describing and exchanging information and handling transactions within companies. Web Services can also fulfill external integration needs for organizations by allowing partners or customers secure access to information residing on multiple internal systems. Web Services ease the challenge of data communication among partners by moving integration from proprietary, point-to-point connections to standards-based connections that can leverage transformation capabilities in various tools.

For organizations who have yet to consider SOA, the primary business application of Web Services is often to improve integration.

For organizations who have yet to consider SOA, the primary business application of Web Services is often to improve integration. Typical enterprises should expect to reduce integration costs by a reasonable amount, considering the immaturity of the solutions available at this time. However, the real business value from Web Services lies in their longer-term potential. As the various components of the Web Services model fall into place and enterprises learn how to best take advantage of SOA, Web Services technologies will continue to save money by simplifying integration.

Reducing dependency on expensive middleware

One of the mantras of the post-dot.com decade is that companies must continue to cut the cost of IT and lower their overall spending. Companies are now stuck in the single-minded thinking that such spending now must decrease year over year, posing a challenge to organizations that must not only maintain existing IT systems, but also implement new technologies and approaches, such as the kind that SOA represents. How can such firms ever hope to continue to innovate if all they are doing is reducing their spending year after year? Fortunately, SOA presents the interesting paradox: even though architecture requires new spending, it leads to the reduction of other IT spending to a much greater degree.

In the enterprise environment, the cost of integrating disparate systems can take up much of a project's budget. Point-to-point integration, which consists essentially of hardwiring two systems together, can lead to exceptionally high total cost of ownership (TCO) because of the number of connections needed in a multiple-system environment. As a result, monolithic integration architectures typically depend on a bus or hub and spoke approach, where every system plugs into a common infrastructure, and the bus handles the translations necessary for each system to talk to the others.

Such monolithic solutions, however, have long implementation timeframes and do not deliver the desired flexibility. In addition, the labor cost of such approaches often far exceeds the cost of the software, because they are typically proprietary and complex and create an infrastructure that fails to deliver flexibility and adaptability. Clearly, there is a great demand for a more flexible and cost-effective approach to integration.

In particular, SOA enables the reduction or elimination of tightly coupled middleware, or at the least, a lessening on the dependency on middleware and its associated maintenance costs. As companies solve their integration challenges architecturally, there will be less need to use proprietary, bulky and inflexible middleware. Furthermore, the cost of architecture is significantly less than most firms are spending annually on inflexible middleware solutions.

This cost reduction from SOA is more than simply the elimination of software licenses. The more middleware that companies can get rid of, the fewer hardware systems, consulting services, and even internal staff are needed to manage the remaining complexity. And for those middleware solutions that organizations are unable to replace, SOA helps to abstract their capabilities, enabling organizations to incorporate older middleware into newer infrastructures in a non-invasive way. As a result, the enterprise may be able to scale back any work on the older software, and position it for eventual replacement.

Obtaining more value from legacy

While many organizations can achieve the cost benefit of reducing their dependence on expensive middleware, often the older systems and applications in the IT environment still provide essential business value. Such systems and applications, known as legacy, come in many forms: custom-coded applications with long-lost source code, unsupported packaged applications, or mainframe-based programs with proprietary interfaces. The paradox of legacy is that for most enterprises, these systems are too old to justify ongoing improvements, but are nevertheless too important to retire. As a result, companies often feel that they must continue to maintain their investments even though the returns they receive from these older systems may diminish over time.

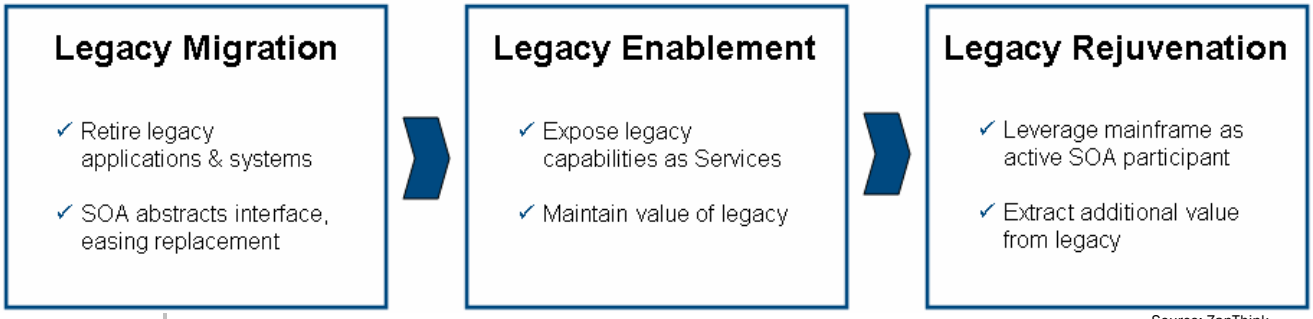
Actually, legacy systems wouldn't be such a cause for consternation, if it weren't for the fact that so much business value resides on these systems—both in the form of essential data as well as critical business logic. Migrating these complex systems is costly and time consuming, and in cases where the mainframe is mission critical to the business, is simply not an option for some organizations. After all, if it were easy to retire these systems, there wouldn't be nearly as many of them around.

Retiring such systems—the dreaded "rip and replace"—is a high-risk and costly task. As a result, organizations must make the best of IT environments with a wide mix of different legacy systems. Indeed, simply maintaining existing legacy systems is less risky than replacing them with entirely new applications fraught with perils of their own. Organizations are far more willing to spend money on extending their legacy systems by incorporating them into the SOA world by insuring the reliability of the legacy system while exposing its functionality and data as composable Services.

Many technologists have found it difficult to build or run new applications on legacy platforms, and many times older operating systems, enterprise applications, or middleware fail when made to perform new tasks they were not originally designed for. There's no denying that in some cases, the legacy system poses significant problems that lead to reduced business value because the software no longer meets business needs or is too inflexible to change. Some organizations also find themselves in the situation where certain legacy systems duplicate existing functionality, either with other legacy or with newer systems or applications. In those cases, they will likely seek to develop a legacy migration and retirement strategy.

In other cases, the business imperative is to keep the legacy applications and gain as much value from them as possible, which is where SOA can help. Extracting such value out of legacy has many facets. The adoption of SOA can simplify the process of legacy migration, as well as legacy enablement and legacy rejuvenation. An illustration of the relationship among legacy migration, enablement, and rejuvenation within the context of SOA is shown in the figure below.

Increasing Value of Legacy within SOA



Source: ZapThink

Legacy migration, while it can reduce the ongoing costs of maintenance, can also present substantial risks to the enterprise. Rarely is an older system entirely superfluous; far more often it still serves some valuable purpose at the time of its retirement. In those situations, it is essential to transition the consumption of its capabilities to a replacement system in as seamless a way as possible. More frequently, however, organizations find that they don't actually require the retirement of legacy systems. Instead, they face issues relating to the usability of the legacy applications, or the applicability of those applications to evolving business needs.

In such cases, it makes more sense to put together a legacy enablement strategy that seeks to improve the value of the system in question without requiring its retirement. Web Services provide the flexibility to abstract mainframe functionality for immediate reuse or as a path for easier migration. For example, one migration scenario would be to wrap legacy transactions as a Web Service and in turn expose the Service via a Web interface, supporting an online application. It might also be feasible to migrate a duplicate instance of the legacy Web Service implementation to a distributed environment, enabling the eventual decommissioning of the original legacy transaction without any disruption to the business. Achieving the value of legacy enablement, however, requires careful cataloging and documentation of artifacts associated with those legacy capabilities.

It's necessary to rejuvenate the legacy environment to obtain more value out of older applications than their original programmers intended.

Such legacy enablement can clearly provide new business value, but in today's environment of rapid change, increased competition, and limited budgets, the business is calling upon IT to squeeze even more value out of legacy technology than migration or even enablement can provide. To meet today's IT challenges, it's necessary to rejuvenate the legacy environment, in other words, to obtain more value out of older applications than their original programmers intended. The greatest challenge to legacy rejuvenation involves making changes to the legacy environment itself; in fact, in many cases, such invasive changes are impractical or impossible. The secret to successful legacy rejuvenation,

therefore, is to employ a non-invasive approach that extracts new value out of older systems without requiring risky changes to the systems themselves.

III. Achieving Asset Reuse

Extending the value of existing legacy applications is in essence a form of asset reuse. For many organizations, however, reuse is one of the primary motivations for SOA, regardless of whether they plan to reuse any applications they have identified as legacy. Fundamentally, SOA promises the elimination of redundancy through increased reuse of IT assets—essentially providing Return on Asset (ROA) instead of simple ROI. Such redundancy often results from the fact that companies have spent hundreds of millions of dollars implementing the same basic projects over and over for different purposes.

For example, most companies implemented their first customer databases a few decades ago, but they are still implementing the same sort of application functionality to this day in their CRM, ERP, portal, and Web-based systems. When will all this redundant and unnecessary spending end? Only when we can get true reusability of Services. SOA enables companies to build Services once and reuse them multiple times to solve different challenges as they arise. Reuse has always been challenging, but is one of the most important benefits of SOA.

Reducing redundancy

Eliminating redundant capabilities by sharing Services is perhaps the obvious reuse benefit of SOA. Yet, reuse has long been a goal of software development, and has been more difficult to achieve than people have expected it to be. Perhaps the real problem of reuse is that building for reuse means building an asset that is by definition beyond its immediate and specific business requirements. How can a developer ever hope to build a reusable asset if the specifications for reuse are by definition unclear, imprecise, and apt to change frequently? Is reuse even possible if it's impossible to define the specific functionality for a reusable asset? And if a developer must create some reusable code, how will he or she know when it is done?

In many ways, reuse is not in the eye of the person building the asset, but rather in the eye of the people who use it. A developer can code thousands of versions of an asset for every possible scenario and foreseen consequence, but find that people use only a few of them. On the flip side, the developer might find that no one is using those assets at all because, despite all the different variations, they still don't meet a particular and specific need. So, the real challenge is to make reuse a reality in the eye of the user rather than in that of the developer.

The new mantra for reuse in the Service-oriented context is the notion of broad applicability.

The new mantra for reuse in the Service-oriented context is the notion of broad applicability. Instead of sharing code, Service consumers share Services once they're up and running. Clearly, we aren't multiplying a Service for different purposes by copying and pasting it multiple times, and neither are we using a Service in the same way as code or linked libraries. Mainframe systems, application servers, or data sources might provide Services for that matter, and there's no dependence on the particular choice of underlying infrastructure, which removes one of the problems that libraries faced. Instead, by putting the variable aspects of a Service in the Service contract, rather than in the underlying code, we can build a shared Service that is broadly applicable to many different processes and consumers. Such reuse requires careful cataloging of assets as well as analysis of the dependencies across Service implementations.

If a Service is responsible for providing customer data, for example, but a dozen different consumers want the Service to present those data in different ways, we're presented instantly with a reuse conundrum: Should we make a dozen different Services, a dozen Service contracts, or a single contract with enough variability to handle the disparate needs? In the first case, we wouldn't want to create a dozen different Services, because that would defeat the goal of reuse. A single, overly complex contract would not be agile. A dozen different contracts for a single Service is a lot more manageable, and agile as well. Instead of managing a dozen Services, we now have to manage a dozen contracts, but that is a significantly easier task. Broad applicability, therefore, depends on the management of Service contracts.

Building broadly applicable Services doesn't guarantee reuse if people can't find those Services or understand how to use them.

Building broadly applicable Services, however, still doesn't guarantee reuse if people can't find those Services or understand how to use them. In addition to being broadly applicable, therefore, Services must also be consumable. Consumability means that people actually are able to reuse the Services. There must be a registry or repository that serves as a clearinghouse for available Services. Users must know how to use that clearinghouse to find the Services they need. Consumability also requires that there be sufficient information about the Services so that people know which Services are the right ones, and how to actually use them—and yes, this additional information is more Service metadata that the company must manage.

The Holy Grail of reuse is therefore finally within reach, because SOA takes a practical approach to reuse. It's human nature to reuse something only when it's flexible and easy to find and use. By building Services to be broadly applicable and consumable, we're making it easy for people to reuse them.

Improving customer value

While enabling Service reuse to reduce redundancy is the most obvious reason to focus on reuse as a SOA value proposition, it is not the only one. In fact, many organizations strive for reuse for a better customer experience. Reusable customer-facing Services and customer support Services both improve the customer experience and increase the value of the organizations most important customers. Reusable customer-facing Services and customer support Services both improve the customer experience and increase the value of the organizations most important customers.

Take, for example, a large retail bank, who offers deposit accounts, credit cards, brokerage services, and treasury management services to its customers. Such large banks typically operate as siloed organizations, especially when they have grown through acquisitions. The typical customer experience when calling the bank's call center, say, is to get information on one type of account. But if a customer calls about their credit card and then asks about their bank balance, the call center typically cannot respond effectively, because the requisite systems and processes are in different silos.

This breakdown in customer service is a particular challenge to the banks and other large enterprises with similar problems because the enterprise's best customers tend to be the ones with accounts in multiple silos. So, while it makes business sense to treat their top customers best, just the opposite is taking place. Fortunately, SOA can help solve this problem.

What many large banks and other enterprises are doing is building core sets of shared Services that can support their customers across their silos. These shared Services tend to fall into customer-related Service domains, as they can provide the call center, interactive voice response (IVR), the Web site, and the

branches with common access to Services that provide the requisite information and capabilities that the organization needs to keep their best customers happy.

In addition to the technical challenges that this application of shared Services introduces, it is also requires dealing with the collaboration challenge between Service providers and consumers. Service domains consist of a set of Services that share a common business context. As a result, the Service domain must serve multiple lines of business, and also interact with various IT silos. In essence, then, building and running Service domains is a collaboration challenge more so than a technical one.

IV. Increasing Business Visibility

The complexity of today's heterogeneous environments hides critical information about the organization, its customers, and its business. Many organizations leverage SOA for no other reason than to improve business visibility in this kind of environment. In some cases, this value proposition overlaps the reuse value proposition, since improved customer value depends on improved business visibility. It's of value to discuss these benefits separately, however, because they are very different business motivations for many organizations.

Regulatory compliance

Regulatory compliance is a mandate that no organization can ignore. Executives' freedom may hinge on having sufficient visibility into the organization's finances or other aspects of the business. IT organizations must establish mechanisms for more effective, systematic control of certain business processes, even though compliance issues often cut across national boundaries. An increasing focus among enterprises on transparency, reporting and risk mitigation indicates that the demand for compliance capabilities will continue to explode.

Executives must spend whatever it takes to become compliant—but generally not a penny more.

Regulatory compliance is an unusual business motivator for a few reasons. First, executives must spend whatever it takes to become compliant—but generally not a penny more. Secondly, compliance calls for constant attention, tweaking and vigilance as well as a balance of cost, risk and transparency. Furthermore, as regulations are arbitrary and might change at any time, organizations must be able to respond to such changes in an agile manner. For all these reasons, organizations are turning to SOA to help them with their regulatory compliance challenges.

Compliance projects can face substantial technical challenges. Despite the increasing attention on compliance as a pervasive business concern, technical efforts to address the various challenges that compliance poses requirements often lose out to more tactical initiatives. In many cases, compliance initiatives overlap, say when an organization merges their Sarbanes Oxley project and Basel II project, even though the respective implementation teams may have little knowledge of each other's activities.

As a result, many organizations find that overlapping point applications require integration with the organization's newer compliance applications. Without SOA, addressing specific tactical challenges yields a complex, redundant infrastructure which replicates functionality while producing both higher initial costs as well as additional ongoing systems management headaches. Organizations are finding that building a dedicated Basel II compliance application, for example, may not be capable of scaling or responding to the inevitable changes to the Basel II regulations. Another set of pitfalls result from line-of-business executives who operate in divisional or departmental silos, and

don't take into account how compliance issues affect the rest of the organization or how IT might be able to help the enterprise overall with its compliance needs.

By leveraging SOA best practices to pull together disparate compliance requirements and distil a core set of Services, organizations can focus on creating compliance-specific Services. In this way, it's possible to incorporate compliance with existing SOA efforts. Leveraging SOA for compliance brings IT and business goals in line. It also addresses integration challenges and resolves the issues that departmental silos can introduce.

In fact, SOA simplifies and streamlines compliance. Incorporating compliance Services in an SOA initiative can save time and expense. When IT organizations apply the benefits of SOA apply to compliance initiatives, they are able to reduce the number of redundant purchases, resulting in lower licensing fees, increase their productivity from Service reuse, improve their time to market for Services, improve management and flexibility necessary to adapt to changing regulations.

Not only must SOA initiatives pull together heterogeneous data in order to support regulatory compliance initiatives, but the business also requires a way to measure the success of such initiatives. While the measurement of success is vital to achieving any business benefit, it is central to the benefit of compliance, because it is never sufficient for an organization to be compliant—they must *know* they are compliant. Visibility and measurement, therefore, are critical SOA capabilities for the benefit of regulatory compliance.

Business efficiency

While regulatory compliance is often the most urgent priority, simply running the business efficiently is an ongoing challenge for the executive team, and requires seamless collaboration between business and IT. While there is nothing restricting compliance to preventing negative corporate behaviors, rarely do CIOs pursue the tangible business benefits of complying with regulations. Nevertheless, a few progressive CIOs are looking for gains in business quality from compliance initiatives like their Sarbanes Oxley efforts. Basel II compliance, furthermore, frees up capital that reduces financial risks, which is helping to motivate companies to comply with this international regulation.

Rarely does an executive team put "business efficiency" line items into their budget.

And yet, rarely does an executive team put "business efficiency" line items into their budget. Instead, a focus on business efficiency generally takes place when there are particular examples of inefficiency that cause the organization some form of pain. Inefficient business processes that lead to particular problems, therefore, are a key business motivator for SOA. The sorts of problems that fall into this discussion include processes that take too long or cost too much to execute, processes that lower customer satisfaction, or processes that are not flexible enough to respond to changes in the business environment. In all of these cases, leveraging SOA can address the issues at hand.

The key to improving business efficiency with SOA, therefore, centers on compositions of Services that result in flexible, composite applications called *Service-Oriented Business Applications* (SOBAs) that implement business processes. SOBAs provide for the flexibility and control over business processes that organizations require to improve the processes' efficiency.

SOBAs enable companies to improve their business processes in several ways. Clearly, SOBAs must provide a range of business process management capabilities, including the ability to handle well-defined orchestrations that consist of sequences of process steps with predefined flows, including Web Services Business Process Execution Language (BPEL) flows, as well as more *ad hoc* processes defined as choreographies that only specify preconditions and

postconditions for interaction in a non-programmatic manner. SOBAs must also maintain an accurate representation of the processes in production, even as they undergo change.

Furthermore, SOBAs can offer the visibility that existing dashboards currently provide. SOBAs can bring dashboards that present siloed data in a siloed manner together, provide new approaches for combining the knowledge each one provides, and enable various people within the organization to take action on the information they glean from this improved dashboard, all contributing to the ability for the organization to improve their business processes in an agile manner.

Risk mitigation

Risks to the organization come in many forms: fraud, security breaches, failed projects, and in general, responding ineffectively to business change. To both reduce the risks of SOA as well as to leverage SOA for better risk mitigation overall, organizations require both change management and measurement of the success of their SOA initiatives. An architectural approach like SOA is necessary to improve the alignment of technology with business goals and manage risk within the organization.

Specifically, SOA primarily provides a way to significantly reduce risk by providing increased operations visibility. Governance, compliance, and general risk reduction offer a different quantifiable benefit than increased business agility. Compliance and governance lead to a reduction of liability, while business agility offers an increase in business opportunity. All are important, but they speak to different parts of the corporate psyche. Just how much is compliance worth? The answer lies in how much noncompliance will cost a company.

Senior management can no longer ignore IT governance and compliance issues. Enterprise architecture provides an effective framework for addressing statutory and corporate governance requirements by improving planning, providing the ability to prove compliance, increasing executive management's visibility and control, and offering a better understanding of the value of technology investments.

V. Achieving Business Agility

While cost reduction, asset reuse, regulatory compliance, and business efficiency are all critical motivations for SOA, business agility is perhaps the most important business benefit of SOA overall. And yet, of all the benefits SOA offers, business agility is perhaps the most difficult to calculate an ROI for. For this reason, many organizations focus on the other value propositions described above as a way to justify a SOA initiative, at least to build acceptance for SOA in their organizations. Only after showing a level of success with SOA will business agility be a primary motivation for additional SOA iterations.

Only after showing a level of success with SOA will business agility be a primary motivation for additional SOA iterations.

To understand the business agility benefit, it's important to note that the definition of business agility has two parts. ZapThink defines business agility as being able to respond quickly and efficiently to change, and leverage change for competitive advantage. The first part of this definition, responding to change, is the more reactive or tactical part of the definition. The second part, leveraging change for competitive advantage, is more proactive and strategic. In fact, leveraging change for competitive advantage is the essence of innovation in the enterprise. As a result, business agility and innovation are closely related.

Time to value

Sometimes, the focus of innovation is in bringing new, unique products to market. In other instances, innovation focuses on bringing new products to market *faster.* In some industries like telecom, there is continual competitive pressure to bring out new products and services. If one enterprise can bring out a new product one or two months faster than the competition, that time to value advantage goes straight to the bottom line.

This business benefit clearly overlaps with business efficiency, although in this case, the efficiency the organization seeks specifically relates to rolling out new products and services. SOA can be a critical enabler of such efficiency by streamlining the processes for bringing products to market, and also by rationalizing the legacy infrastructure that goes into many of these converged products.

For the telcos, convergence is now a way of life, as they try to pull together wireline, wireless, cable television, and other capabilities to offer products that cross these silos, but telco isn't alone in this respect. Any industry who finds that innovation centers on crossing existing silos can benefit from the time to value advantages of SOA, including insurance, banking, travel, supply chain industries, and more.

Keep in mind that time to value doesn't end when a product reaches the market. Clearly, those products and services must continue to provide value throughout their lifetime, and must remain flexible in the face of ongoing innovation. It is critical, therefore, for IT to provide for full lifecycle management of their SOA initiatives in order to continue to guarantee improved time to value for the business.

It is critical for IT to provide for full life-cycle management of their SOA initiatives.

Strategic differentiation

The flip side to the time to value benefit of business agility is strategic differentiation. Recall that our definition of business agility had two parts: Respond quickly and efficiently to business change, and leverage that change for competitive advantage. It is the second part of the agility benefit that promises, but does not automatically provide a strategic differentiator.

To understand how SOA can truly provide a strategic differentiator, let's look at an example. Let's say that sometime in the future a telecommunications company's strategy includes developing a create-your-own-service product offering for their customers. Through the power of SOA, it enables its customers to pick and choose from a range of sophisticated telecom capabilities and assemble them into a new service unique to each customer. Furthermore, customers are able to rework this service any time they like. So, if a customer thinks it would be a good idea that whenever a particular show came onto the telecom-provided digital television service, the TV launched a multimedia conference call that brought together a group of the customer's friends into an online, interactive viewing experience via their mobile phone/personal digital assistants (PDAs), then the customer can simply go to the telecom company's Web site and set such a service up.

Such an offering would be strategic if it uniquely differentiated the company from its competitors. And there's no way a company could offer such a dynamic, continually changing service if it wasn't Service-oriented. Furthermore, the power of this service is that customers can take it wherever they want to go—essentially, the marketplace itself determines the nature of the product on a day-to-day basis. The whims of the marketplace, after all, are among the most fundamental sources of unpredictable business change. This telecom, therefore,

is using the power of SOA to leverage such change for competitive, strategic advantage.

SOA alone cannot give a company a strategy.

It's important to point out, however, that SOA doesn't necessarily offer such differentiation. Companies frequently use SOA to leverage changes in the business environment solely for tactical competitive advantages, such as bringing a better product to market. SOA alone, after all, cannot give a company a strategy. It is all too easy for SOA to fall into the trap of simply being another approach that IT uses to meet tactical business needs. Many of the benefits of SOA we discussed above fall into this category of tactical improvements: reducing the cost of integration, Service reuse, and enabling regulatory compliance, for example. Those organizations who are able to leverage SOA for strategic advantage, however, are better able to achieve their long-term business goals.

Dealing with unexpected change

Both the time to value and strategic differentiation aspects of the business agility benefit both depend upon astute business planning. But not all business eventualities are part of a plan. Whether it be natural disaster, a hostile takeover, or some sort of political or economic upheaval, businesses must occasionally respond to unexpected changes in their environment.

Unexpected changes in the marketplace are by definition impossible to predict, and thus very difficult to plan for. To achieve the agility necessary to respond to such changes requires effective change management and adaptability of solutions to the needs of the organization. SOA alone can help with such levels of agility, but are by no means sufficient. In fact, resilience is an inherently human trait, and some organizations simply have more than others. The fact remains, however, that a resilient organization that has SOA in place is much better prepared for unexpected change than an equally resilient firm with less flexible IT.

A resilient organization that has SOA in place is much better prepared for unexpected change than an equally resilient firm with less flexible IT.

VI. Software AG CentraSite: Achieving the Benefits of SOA through Governance

So far, this paper has carefully delineated the business benefits of SOA, and yet, we haven't focused on the area of SOA Governance. This emphasis on business benefits, of course, was intentional; the point to the paper is to encourage enterprises who are planning or implementing SOA to begin with the business problems and identify the appropriate solutions for them.

That being said, somewhere the rubber must hit the road—once architects have identified the appropriate solution to a particular problem, and that solution falls under the auspices of their SOA initiative, the next step is to insure that their SOA planning and implementation efforts actually address the problem they intended to solve. That's where SOA Governance comes in.

ZapThink has written many White Papers on SOA Governance, and in any case, an in-depth treatment of the topic is not within the scope of this paper. Instead, we're taking the capabilities of the CentraSite SOA Governance solution and showing how it promotes the benefits of SOA described above. It's important to understand that organizations should not select a product like CentraSite based on its features alone. Rather, it's essential to understand how the capabilities of the product address the business challenges that the organization is looking to SOA to solve.

CentraSite's core benefits support the other SOA value propositions that this paper discusses.

Software AG and Fujitsu's CentraSite is a SOA Governance solution built on a powerful, open and standards-based SOA Registry/Repository. CentraSite enables enterprises to achieve control and transparency across all IT assets within the organization and monitor SOA implementations with reporting capabilities.

Fundamentally, however, Software AG approaches SOA Governance as being critical for achieving greater business agility, because achieving the benefit of agility impacts how organizations manage and govern their SOA environments. In fact, CentraSite's core benefits support the other SOA value propositions that this paper discusses. In particular, CentraSite enables organizations to accomplish the following:

- *View IT assets, Services, policies and processes holistically* – CentraSite's ability to provide visibility across heterogeneous metadata and processes aids with regulatory compliance, improves business efficiency, and can also reduce time to value for innovative products and services.
- *Enable rapid and low-risk change management that changing business needs require* – CentraSite's built in change management capabilities aid with regulatory compliance initiatives, improve business efficiency, mitigate risks, lower time to value, increase strategic differentiation, and also enables organizations to deal with unexpected change.
- *Implement SOA based on proven methodologies* – CentraSite incorporates and encourages established SOA best practices, which help companies to leverage Web Services properly, reduce redundancy, and improve the time to value of their new products and services.
- *Measure SOA success with the right business and technical metrics* – Improved visibility is the key to improving customer value, regulatory compliance, business efficiency, and risk mitigation.
- *Improve collaboration between business and IT, as well as between Service providers and consumers* – CentraSite's collaboration capabilities help organizations in many ways, including improving customer value and business efficiency, and can also help with regulatory compliance for regulations that affect multiple silos.
- *Adapt the solution to the organization, rather than the reverse* – The adaptability that CentraSite provides improves customer value and business efficiency, and also aids organizations develop strategic differentiation in the marketplace.
- *Catalogue and document any SOA artifact, analyze dependencies and improve reuse* – CentraSite's versatile cataloging capabilities help organizations obtain more value from legacy, reduce redundancy, and deal with unexpected change.
- *Optimize and manage the full SOA lifecycle for Services, policies and processes* – CentraSite's Service lifecycle management capabilities are critical for lowering time to value and dealing with unexpected change in particular.
- *Integrate with any third-party solution by plugging into the open and standards-based architecture of CentraSite, as well as integrating with over twenty SOA infrastructure vendors and CentraSite community partners, such as AmberPoint, Fujitsu, Ilog, IDS Scheer, Mindreef and Novell, as well as Software AG webMethods* – CentraSite plays far more than a supporting role for Software AG's integration products. Its open

standards support and integration with third party products reduce dependency on expensive middleware and enable organizations to obtain more value from their existing legacy systems.

VII. The ZapThink Take

Whenever an organization innovates, there is always inherent resistance within the organization to the new product or approach. Innovation does not become execution until the stakeholders responsible for delivery adopt the new idea. And yet, people are inherently resistant to such change, not because they are bad people, but rather because they have responsibilities, constraints, and expectations that limit their ability to accept new ideas.

One of the most important roles of governance in the organization is to make such responsibilities explicit rather than implicit. For example, the Corporate Governance group has the responsibility to ensure that regulations are being met. Other groups have other responsibilities that also make up policies that form part of the enterprise's governance framework.

By making such policies explicit and part of an automated system, these objections become less territorial and personal and more of an enforceable, automated way of doing business. As a result, the enterprise governance infrastructure can automatically tell you why a certain course of action is not advisable. This advantage empowers the innovators in the organization to maneuver around roadblocks.

And yet, this innovation benefit of SOA is largely out of reach for most organizations, as enterprises continue to struggle with building the business case for SOA, while many vendors push product features as providing SOA without tying those features to SOA best practices. Will history regard SOA as a promising architectural approach that ended up being little more than a set of software features?

Fortunately, an increasing number of enterprises are coming to understand the true nature of SOA, and for those architects that do see the light, they are able to identify those business problems that SOA is well-suited to solve, and are able to connect those business problems to the SOA best practices they must implement to be successful in addressing those problems. These architects aren't falling for the "SOA as product features" line that some vendors are espousing.

An increasing number of vendors like Software AG are seeing the light as well, and are positioning their products as helping implement SOA best practices. After all, organizations who get the architecture right understand that best practices are essentially forms of human behavior independent of the technology. Once you have those best practices, the products you purchase can help you implement them, but the products will never be the source of the best practices themselves.

CentraSite is a perfect example of a product that not only has features that help organizations implement SOA, but also directly supports a range of SOA best practices as well.

In fact, CentraSite is a perfect example of a product that not only has features that help organizations implement SOA, but also directly supports a range of SOA best practices as well. This paper, therefore, illustrates this point: begin with the business problem, identify SOA best practices, and only then select a product that will help implement them.

Copyright, Trademark Notice, and Statement of Opinion

About ZapThink, LLC

ZapThink is an IT advisory and analysis firm that provides trusted advice and critical insight into the architectural and organizational changes brought about by the movement to XML, Web Services, and Service Orientation. We provide our three target audiences of IT vendors, service providers and end-users a clear roadmap for standards-based, loosely coupled distributed computing – a vision of IT meeting the needs of the agile business.

ZapThink helps its customers in three ways: by helping companies understand IT products and services in the context of Service-Oriented Architecture (SOA) and the vision of Service Orientation, by providing guidance into emerging best practices for Web Services and SOA adoption, and by bringing together all our audiences into a network that provides business value and expertise to each member of the network.

ZapThink provides market intelligence to IT vendors and professional services firms that offer XML and Web Services-based products and services in order to help them understand their competitive landscape, plan their product roadmaps, and communicate their value proposition to their customers within the context of Service Orientation.

ZapThink provides guidance and expertise to professional services firms to help them grow and innovate their services as well as promote their capabilities to end-users and vendors looking to grow their businesses.

ZapThink also provides implementation intelligence to IT users who are seeking guidance and clarity into the best practices for planning and implementing SOA, including how to assemble the available products and services into a coherent plan.

ZapThink's senior analysts are widely regarded as the "go to analysts" for XML, Web Services, and SOA by vendors, end-users, and the press. Respected for their candid, insightful opinions, they are in great demand as speakers, and have presented at conferences and industry events around the world. They are among the most quoted industry analysts in the IT industry. ZapThink was founded in November 2000 and is headquartered in Baltimore, Maryland.

ZAPTHINK CONTACT:
ZapThink, LLC
108 Woodlawn Road
Baltimore, MD 21210
Phone: +1 (781) 207 0203
Fax: +1 (815) 301 3171
info@zapthink.com

ZapThink, LLC • 108 Woodlawn Road • Baltimore, MD 21210 • info@zapthink.com • www.zapthink.com