

# SOA WORST PRACTICES, VOLUME II: A LOOK AT GOVERNANCE

*Copyright ©2006. Progress Software Corporation.  
All rights reserved.*



**PROGRESS**  
SOFTWARE

**MOMENTUM**  
si



---

## TABLE OF CONTENTS

|   |    |
|---|----|
| > Introduction: Cautionary Tales  | 2  |
| > The Seven Worst Practices   | 3  |
| > 7. Governance, as a Process, Can be Checked at the Gates                          | 3  |
| > 6. We'll Do Anything for Our Best Customers: Even Duplicate Our Data Center       | 6  |
| > 5. Divide and Conquer: Approaching Security, Management and Governance Separately | 8  |
| > 4. As Long as I Don't Know About It, Compliance Doesn't Matter                    | 10 |
| > 3. No Worries: We Know Who's Using the Web Services that Come With Our App        | 12 |
| > 2. Hey, This is a Really Cool Registry; Our Web Services Designers Will Love It!  | 14 |
| > 1. No Thanks: We Have Our Own Standards and Processes Here                        | 17 |
| > Parting Thoughts  | 19 |
| > For More information  | 20 |

---

## INTRODUCTION: CAUTIONARY TALES

SOA Worst Practices, Volume II, as the title suggests, is the follow-on white paper to SOA Worst Practices, Volume I: a collection, some readers may recall, of dubious—and sometimes disastrous—case studies of service-oriented architecture (SOA) implementations. Each scenario describes and analyzes the critical SOA errors and lessons learned, and then a prescription is revealed to stimulate discussion and corrective possibilities.

### A GOVERNANCE FOCUS

The first SOA Worst Practices white paper from Progress Software focused on tactical and strategic aspects of implementing service-oriented architectures: employing SOAP standards, the limits of XML firewalls, Web-services reuse and so on. This paper, by contrast, will zero in on the worst practices of organizations that attempted—or attempted to avoid—SOA governance.

### *SOA Juggernaut*

SOA fever is reaching peak levels in the IT industry and in the IT-industry media: there's simply no escaping it as company after company seeks to squeeze competitive advantage from its information infrastructure. Now, even the major packaged-application vendors are delivering built-in Web services with their applications—forcing IT organizations that would otherwise have wished to put SOA on the back burner to face the prospects of their SOA enterprise initiatives today. For these organizations, a key question arises: *how should you manage and govern these new services as they are being deployed on your network?*

### *Terms of Confusion*

While there is a lot of excitement surrounding SOA technology, there is still great confusion over the terms used to describe what is included in SOA. A glance at Google as of this writing shows that *SOA governance* produces nearly 4 million hits, while *Web services management* returns 398 million hits. Given the millions of pages of content published on the Internet, coupled with the sense of relative novelty associated with the subject, it's no wonder different suppliers, service providers, analysts, members of the press and IT managers promote their own, often unique, definitions of the concept. With the prevailing confusion around the term governance, therefore, it's very easy for people who think they are doing the same thing—addressing governance—to be moving, in reality, in opposite directions.

---

## THE SEVEN WORST PRACTICES

The case studies that follow tell the cautionary tales of several organizations<sup>1</sup> whose not-so-best-laid SOA governance plans went awry. These cases include:

- 7. *Governance, as a Process, Can Be “Checked at the Gates”***
- 6. *We’ll Do Anything for Our Best Customers: Even Duplicate Our Data Center***
- 5. *Divide And Conquer: Approaching Security, Management and Governance Separately***
- 4. *As Long As I Don’t Know About It, Compliance Doesn’t Matter***
- 3. *No Worries: We Know Who’s Using the Web Services that Come With Our App***
- 2. *Hey, This is a Really Cool Registry; Our Web Services Designers Will Love It!***
- 1. *No Thanks: We Have Our Own Standards And Processes Here***

Read on to see what happens when different organizations in a variety of fields execute on their own (sometimes unique) vision of SOA governance.

## 7. GOVERNANCE, AS A PROCESS, CAN BE “CHECKED AT THE GATES”

It would seem logical, on its face, that governance mechanisms could be designed, associated with individual services, and implemented upfront—and that IT could then assume that their role in governance is complete. Despite the apparent logic of this governance approach, it almost inevitably leads to trouble as policies and procedures change over time—in unexpected ways.

### THE IDEA

Amalgamated Electronics is a multinational consumer-electronics manufacturing firm. Amalgamated’s IT team, like many others, knew that the success of their SOA would depend on the governance of its Web services. The team spent many hours interviewing individuals in the company to understand the policies that would need to be created. But, the team’s governance focus was too static; there was an assumption that the architects could design in governance up front and that everything in future phases—development, test, deployment and runtime—would remain the same. Simply put, the plan was that governance would work as built. Period.

Clearly Amalgamated’s IT architecture team was confident of its plans. Or perhaps they simply had unusually great faith in the universe’s sense of fair play.

---

1. Although the names of all the companies mentioned in these cases (except for ours) have been changed, their stories are no less true or painful.

---

### **Multiple Hats**

In addition to writing code, it was the developers who were tasked with making sure services complied with defined company and regulatory policies. And it was during the development phase, therefore, that developers built the dependencies between Web services and corresponding policy enforcement (outlined by architects in the design phase) on a service-by-service basis.

*So what was wrong with this approach?*

### **WHY IT WASN'T SO SMART**

The fundamental problem with Amalgamated's approach to Web services governance was that it assumed that nothing on its services network would change. Or, at minimum, it assumed that change would be simple to track. Their applications, as designed, might pass their governance criteria, and work well at the time they handed them over to the development team. But that's where the trouble began.

Amalgamated's IT team had no visibility, beyond how they designed things, into how the governance measures affecting its applications were implemented, tested, deployed—or how they fared in runtime. Nor, of course, did they have any clear notion of the "change" relationship between what was designed and what was running on the network over time. So, while Amalgamated's developers may have "designed in" governance that worked at first—as policies inevitably changed and services that weren't initially designed were deployed, IT found that its design-phase, service-by-service approach proved too brittle a form of governance to last. Soon enough, compliance policies and other forms of governance on the network began to break.

### **Bottom Line**

The Amalgamated IT team should have asked themselves the following key governance questions:

- > How do we know how change will affect what's already there? In other words: it's one thing to design something we know will remain static, but when we have an environment that we know will be subject to change over time, are we able to understand and manage the effect of this change on our deployed SOA—in a reasonably efficient manner?
- > How do we know what we don't know—if we don't know it in the first place? Once past the design phase, how do we know what is going on? Are there checks and validation in place after development, testing and deployment? Do we have visibility into the runtime environment and the ability to ensure that governance policies are enforced?

---

What Amalgamated's IT team didn't see was that designing for SOA governance, then releasing into a changing, ungoverned world is a recipe for disaster. As regulations change, for example, Amalgamated's out-of-date policies and associated enforcement approach placed the company at risk of fines, loss of revenue—even jail time.

## A BETTER APPROACH

Only by approaching governance from the perspective of the entire "Service Lifecycle" can organizations help to ensure success. Success can only be achieved when there is a consistent application of governance rules and policies across *all services*, from the initial design to the actual runtime environment. Without this focus, companies are at risk of not enforcing measures that they intended to, and of not having the visibility to ensure that only services that are in compliance are deployed. Put another way: Web services governance is about minimizing *security and risk* to the enterprise. And non-compliance with corporate policies as well as various government regulations—including Sarbanes-Oxley, HIPAA and others—represents some of the greatest of these risks.

Complementing design-time governance with runtime governance offers the user a "layer of abstraction" between the services themselves and the governance mechanism: centralizing management of policies and processes so they can be controlled automatically. It is this centralization that permits the automation of policy enforcement and which prohibits the violation of compliance "behind the backs" of business and IT. In addition, several related concepts and procedures need to be in force:

- > Policy enforcement should be decoupled from actual application development and deployment.
- > Policies should be independent of applications; they can and should change independently.
- > Separate roles and responsibilities should exist for policy creation and enforcement: application developers are not policy experts, and can't develop application logic efficiently when burdened with this task.

---

## 6. WE'LL DO ANYTHING FOR OUR BEST CUSTOMERS: EVEN DUPLICATE OUR DATA CENTER

The era of mass marketing is long gone. Employing their IT systems, companies are seeking new and ingenious ways to target and segment their markets and customer bases so they can please these groups—and ultimately extract more profit from their sales and service operations. In this atmosphere, some organizations are tempted to simply build out extra capacity to meet the needs of their most profitable customer segments; while this may seem a sensible strategy at first glance, it does not hold up under close scrutiny.

### THE IDEA

Revere Travel Associates (RTA), a global travel agency, built its original business in the corporate market, with many of its agents working onsite with large client firms such as insurance and financial companies. In recent years, RTA expanded its services to include online ticket booking for its clients. RTA's clients saw the combination of personal service (an agent just down the hall), the convenience of online ticketing—and RTA's ability to consolidate billing and reporting across all of its clients' locations—as major “wins” for them. Nonetheless, the travel business is extremely competitive, and when some of RTA's most important clients began to complain about the performance of their online portals, senior management took the complaints immediately to heart. RTA's CEO gave the IT department a mandate: find a way to improve the user experience of RTA's “best” customers—who also happened to be the company's most profitable.

The strategy IT came up with (working closely with the Sales department) was fairly straightforward: categorize all RTA clients—based on the amount of bookings they committed to in advance—as either Platinum, Gold or Silver customers. RTA would employ its IT infrastructure to guarantee optimum levels of network and application performance to its best, most-profitable clients: namely, its Platinum customers. In effect, RTA was offering and promoting strict service-level agreements (SLAs) to its Platinum customers. The RTA team felt sure this would help secure Platinum customer loyalty, as the key complaint was the poor performance of RTA's ticketing services at peak times of the day.

### *How to Guarantee Performance?*

The trick, of course, was how to guarantee the promised performance. After considering a variety of approaches, the “most obvious” method was settled on: IT would simply add capacity to its infrastructure and dedicate it to Platinum-related service traffic. RTA ended up expanding its data center and purchasing more server hardware and related software. In so doing, IT duplicated much of its existing infrastructure. In this way, the company believed that it would be able to guarantee the service-level requirements of their Platinum customers and keep their top customers away from hungry competitors.

---

## WHY IT WASN'T SO SMART

While RTA's approach would—and did—work, it was less than ideal. Dedicating an entire IT environment (consider all the hardware, software, maintenance, energy and real estate factors!) to meet the needs of a single customer segment was far from cost effective. And it wasn't a scalable solution in the long term. What happens when the company identifies another customer segment that it wants to treat in a special way? ... And another? Does it just continue along this costly, duplicative path? Moreover, this additional capacity was really only necessary during peak-usage times, when the potential to violate SLAs was a reality. At other times, the company wasn't utilizing more than 50% of their entire capacity.

## A BETTER APPROACH

Today, SOA-enabled businesses are bound by service-level agreements to deliver pre-determined levels of application performance and availability to their customers. In many cases, not all customers are "created equal"; service levels are driven by individual business rules that define their relationship with the service provider. In these instances, it is not only an operational issue for IT, but also a key concern for the business owners.

Ideally, RTA ought to have employed a combination of governance measures that would have allowed them to design for and operate their runtime SOA to meet the performance needs of its Platinum customers in a differentiated fashion from its other, say "Silver", customers who may have had to wait a bit longer for service during peak hours.

With a runtime governance solution deployed, RTA could monitor their SOA for potential performance issues and use governance policies that reflect their designed business rules to re-route their Platinum customers to a set of standby services only during the detected peak-load periods. (Instead of building an entire dedicated infrastructure to service these customers—an infrastructure that would remain underutilized most of the time). The combination of well-defined governance policies and a much smaller scale of additional capacity for simply hosting the peak-load standby services presents a much more cost-effective solution.

---

## 5. DIVIDE AND CONQUER: APPROACHING SOA SECURITY, MANAGEMENT AND GOVERNANCE SEPARATELY

Company after company today is discovering the secrets of service-oriented architectures: how SOA increase IT agility, cut process costs, boost revenues and drive more dollars to the bottom line. In this enthusiastic environment, some companies are trying to take the “fast track” to SOA. We’ll see what happened when one company, in order to leverage a SOA in a highly competitive industry segment, decided to approach SOA by dividing the work into three, parallel task forces. After all, this approach would certainly get their SOA “to market” faster, wouldn’t it?

### THE IDEA

BlueSky Tech, Inc., a designer of wafer-handling systems for semiconductor OEMs, operates in a highly competitive market. In order to maintain their industry-leading market share, they looked optimistically toward a SOA to ease their ability to adapt and improve the computer systems and applications used to order and customize their wafer-handling systems to particular OEMs’ specifications.

### *Competition Drives Need for Business Agility*

BlueSky’s IT team had heard a lot about the advantages of service-oriented architecture. And while at first they didn’t know much about the intricacies of SOA, they definitely understood the details of their own business—and why it seemed to be driving them toward the adoption of SOA. Their competition was gaining momentum and they felt the need to operate in a more agile fashion to avoid being overtaken.

So, when the subject of SOA was initially taken up within IT team meetings, it was greeted with enthusiasm. IT management had done its research and had come to the conclusion that SOA was the most effective path toward IT agility and alignment with business needs. They were so optimistic that a move to a SOA needed to happen sooner rather than later, that they began to brainstorm about how they could get there as quickly as possible.

Based on their initial research, senior IT management decided that one of the most critical components to their success in moving to a SOA was having a plan for SOA governance.

### *Problem Solving: The Harvard Business School Approach?*

The next (obvious) step, once the decision to engage in planning and architecting BlueSky’s SOA was approved, was to decide how to approach the overall project. Of particular interest was how the BlueSky IT team would approach their SOA governance plan.

---

In the end, the IT director took what—on its face—might seem an entirely logical approach: one that is sometimes associated with the Harvard Business School (though it's unlikely HBS has written up this case!). Faced with a complex problem, BlueSky's IT director did what many an IT director in her place has done in the past: she broke up the problem into logical, constituent parts – and formed corresponding task forces to solve the resulting smaller problems.

For the governance plan, she created task teams with the following missions:

- > SOA Security: Focus on who has permissions to access services, how the services could be used, and the data-integrity issues associated with those services
- > SOA Management: Determine how to manage services in the runtime environment and provide service-level agreements (SLAs) to users
- > Governance Policies and Processes: Focus on capturing the design-time element of the SOA

With three teams working on these problem subsets, the IT organization could arrive more rapidly at a more effective overall governance solution. Sound sensible?

## WHY IT WASN'T SO SMART

BlueSky's IT team was thinking in the right direction when it identified the various issues to be addressed as part of governing a SOA. That said, even though it might seem logical to split up responsibilities and plans into individual components—especially as a time-saver to speed up the adoption of SOA—this generally doesn't work as well as one might expect.

When people think about building a SOA—if they consider such things as security, management and design-time governance policy separately—they will inevitably find, too late (after the individual analyses are done), that each of these areas will often overlap and impact others in unexpected ways. While the design-time and runtime components of governance are both essential, if they are not well synchronized (as was the case with BlueSky) governance efforts will fail.

The following examples serve to illustrate the points above:

- > BlueSky's security task force devised what they thought were the comprehensive security policies and systems required to secure and control access to their ordering applications. What they didn't focus on (which the governance task force did—given the access to corporate policy information they had) was that BlueSky's clients in the defense industry interact regularly with the Federal government and have more stringent security requirements than their other clients and partners. By the time this fact was accounted for, the security architecture had already been essentially designed—and additional work was required to accommodate this requirement.

---

> The task force implementing BlueSky's approach to managing the runtime aspects of the SOA planned to support the provisioning of SLAs to the users of BlueSky's ordering systems. What they didn't know about, however, was the existence of a corporate policy that imposed larger fines if the SLAs for certain categories of customer and partner orders were violated during the last week of each quarter. Additional standby services and intelligent routing policies needed to be in place to support this requirement, not to mention additional servers that hadn't been ordered. If the efforts of this group had been more closely aligned with the governance policy group, this setback might have been avoided.

### **A BETTER APPROACH**

It's important to take a top-down approach to SOA-governance design and implementation. Realize, for example, that approaches to security, privacy and even key business requirements most likely will be connected—and could impact each other when it comes time to enforce them at runtime.

The bottom-line recommendation here is not to avoid task forces when planning your SOA. The point, instead, is to take a broad, enterprise view of critical topics such as SOA governance, while making sure task forces meet regularly to exchange ideas and remain in synch. In this way, the ultimate governance solution you deploy will be a functioning, unified whole.

## **4. AS LONG AS I DON'T KNOW ABOUT IT, COMPLIANCE DOESN'T MATTER**

In the past—half a decade ago, even—IT organizations could afford to be less responsive to the world around them. The demands from business units for instantaneous analysis of applications and changing rules and policies simply didn't exist. And the regulatory environment—which now imposes sometimes burdensome compliance demands on IT—has changed dramatically, forcing organizations to move with ever-greater speed and attention to processes and policies.

### **THE IDEA**

Trafalgar Retail Stores is a profitable regional chain of department stores. Recently, however, the company's IT team found itself in some hot water because some services deployed on its network weren't taking into account the latest regulations.

With regulation on the rise in the areas of financial reporting, health care, privacy, environmental policy and countless other fields, firms are being pressured as never before to comply with a dizzying array of complex rules. And there's no end in sight. One would think this regulatory onslaught would have placed the creation of a strategic Web-services compliance plan at the top

---

of the list for Trafalgar Retail Stores. Yet Trafalgar chose to take an old-fashioned, reactive approach to managing its compliance issues—updating their compliance only with the next scheduled upgrade of services.

### **WHY IT WASN'T SO SMART**

As part of Trafalgar's aggressive e-commerce initiatives, the company had been allowing customers to apply for store credit cards online. This application process required the firm to gather sensitive customer information—including Social Security numbers. Recently, evolving privacy regulation began to require that all Social Security numbers captured online must be encrypted—and even though Trafalgar's IT team was aware of this regulation—they elected to put off the implementation of the Social-Security encryption algorithm until the scheduled versioning of their services at the end of the quarter.

In the interim, hackers managed to gain illicit access into a production database, where they were able to download some of the unencrypted Social Security numbers, along with customers' associated personal data. The resulting cases of identity theft not only hurt the affected Trafalgar customers, but ultimately impacted Trafalgar as well, which ended up suffering substantial financial losses and bad press resulting from a series of expensive law suits.

Trafalgar found itself out of compliance in this particular case fundamentally because of its "defensive" approach to compliance. The IT team's attitude toward compliance ran something like the following:

- > "We hadn't heard about *that* regulation ..." *or*
- > "We'll just fix those services or applications in our next revision, when and if compliance violations show up"

Essentially, the IT team had a "see-no-evil, hear-no-evil" approach to compliance on its service network.

### **Compliance: No Excuses ...**

The most important thing to realize on this topic is that compliance really does mean compliance. It's not o.k. if you "missed the memo" or really meant to address it. Lack of awareness or a lackadaisical approach to enforcing up-to-date compliance is simply not an excuse—and will not shield your organization from business loss, litigation, fines or even jail terms.

---

## **A BETTER APPROACH**

Adopting a serious stance on governance and the inherent process and technologies that can support it, through the entire service lifecycle, is a requirement in industries where changing governance policies, and a failure to react and update governance measures in a timely fashion, can have serious repercussions. Updating policy “when you get the chance” simply doesn’t cut it in this day and age.

Trafalgar would have been better served by abstracting their compliance policies from the application service logic itself—avoiding the need for time-consuming and cumbersome updates to services each time their policies changed. Deploying a comprehensive, automated SOA management solution—facilitating an active approach to compliance—would have allowed policies to be updated in a seamless fashion, independently of the services they apply to. This approach would have also provided visibility into the runtime environment, ensuring that the most current policies were both in place and being enforced appropriately.

## **3. NO WORRIES: WE KNOW WHO’S USING THE WEB SERVICES THAT COME WITH OUR APP**

Increasingly, ERP, MRP and other enterprise packaged applications are being delivered with built-in Web services. These applications include SAP and Salesforce.com. While these built-in Web service offerings can be a benefit to the organizations that buy these packages, they can present distinct management challenges.

### **THE IDEA**

Not long ago, PC MicroCenters, a regional computer and electronics retail chain, upgraded its SAP software. Like many modern enterprise packaged applications, SAP now features convenient, built-in Web services designed to enhance the utility of the application. These services are popular with end users in large part because they are so easy to employ. As it happened, PC MicroCenters’ IT team was looking forward to employing these services, too.

### ***Service Rollout Plan: Authorization and Capacity Planning***

SAP had always remained under the strict control of IT at PC MicroCenters: in the past, business users typically received the access that IT in general deemed manageable and appropriate. To accomplish this IT used traditional provisioning and authorization mechanisms. This way, IT knew “who was who” and “who was using what”.

But as far as IT was concerned, the new Web services that came with SAP would only be used by an IT-defined set of users (and within IT guidelines). And, perhaps not surprisingly, they took the

---

same approach as in the past—authorizing defined sets of IT users and doing capacity planning based on that narrow group of people.

### **WHY IT WASN'T SO SMART**

Unlike traditional application environments—which can prevent a user from accessing an application entirely, or aspects of an application’s capabilities, based on that user’s identity—in this case, the built-in SAP Web services were entirely visible and accessible to every business user on PC MicroCenter’s network who had an SAP login and password. Unhappily, IT had failed to take this new reality of Web services into consideration.

As it happened, a few of these services proved very helpful (and ultimately, too tempting) to some business analysts in the organization. A number of these analysts found some of the Web services features too enticing to resist. They realized that they now had the ability to do mass downloads of data to Excel (for reviewing weekly sales, shipping, commissions and pricing information). In a number of cases, SAP crashed as a result of the load the analysts were placing on the system—until, finally, IT was able (quite by accident—by means of a hallway discussion) to locate the analysts and put an end to their, albeit well-intentioned, back-channel access.

### **A BETTER APPROACH**

Surveys of IT organizations seem to indicate that many of them would like to simply ignore SOA for now—or at least put it on the back burner. In the past, this might have been a sensible approach for some of them. Indeed, many organizations have no crying need for SOA. But the unavoidable reality is that the big-name packaged applications which are embedded in corporate networks around the globe—SAP, Microsoft, Salesforce.com and the rest—are now all coming out with SOA built in. Which means that, like it or not, IT organizations have to be ready: with a SOA strategy and with a governance approach in mind.

In PC MicroCenter’s case, a comprehensive SOA governance solution could have immediately told IT who was mysteriously consuming those SAP services—and automatically taken action to shut down the unauthorized “rogue Web service” usage before SAP or any other element on the network could have been affected.

---

## **2. HEY, THIS IS A REALLY COOL REGISTRY; OUR WEB SERVICES DESIGNERS WILL LOVE IT!**

With the current popularity of SOA and the advantages of deploying Web services, IT managers are finding it increasingly tempting to buy into new Web services technologies to provide flexibility and agility in their technical infrastructure and additional efficiencies to their jobs. While this approach makes sense from a technical perspective, it can lead IT organizations down a path that may not initially result in the desired outcome—particularly in their relationships with their business partners.

### **THE IDEA**

RayStar Industries is an indirect supplier to the Department of Defense and to private industry of defense and aerospace components. The company has multiple product lines that were assembled over time to a large degree by corporate acquisition. Like many large organizations, RayStar was deploying Web services throughout its various business units and was seeking a way to improve service reuse and simultaneously gain control over the services on its network.

As part of a technology evaluation, one IT team's collective set of eyes fell on a registry. They realized it would help them, particularly in design time, to keep track of services and metadata. In addition, the registry would control the consistency and integrity of new services—by adding a structured approval process to ensure business services were properly reviewed before they were published.

So IT had a cool—and valuable—new tool that would help the team deploy new services with greater control. The team would ultimately be more productive and more agile. And the tool would help the organization to reuse services, thus preserving scarce corporate resources. And everybody would be happy. Right?

### **WHY IT WASN'T SO SMART**

Well, not quite. What gradually ensued on RayStar's network was a muddled proliferation of services, rather than an organized approach to SOA. IT, with the aid of its new registry, began to roll out a large quantity of services—without really understanding their value to the organization as a whole, or their impact on the network. Soon after the registry was in place, the team started experiencing a few hiccups: unexpected service failures, slowdowns in performance of existing applications and other delays associated with bringing new capabilities online.

---

## ***IT and Business Not Aligned***

These technical issues soon became significant political problems, since the roll-out of new Web services was now directly affecting the end users: not surprisingly, business managers grew extremely frustrated over service problems and delays. Rather than achieving the objective of service reuse and control, RayStar had a situation where both services with duplicate functionality were deployed, and where they were being reused in unknown and unintended ways.

The problems got so bad that business management—which had not been involved in the registry purchase decision—began rejecting much of the newly released functionality: in many cases going so far as to demand that their applications were restored to their former modes of operation! The business managers did not focus on how—with an adjustment—these new services might provide them with business value, because they were not in the decision loop from the start and, thus, only saw the negative side.

## ***Limits of the Registry***

While the registry was highly useful in rolling out services, particularly in design-time, its limits stemmed largely from a lack of visibility into the runtime service environment. How were services being used—and by whom? Was reuse being achieved? Were services meeting necessary performance goals? And who was being affected by changing services? The registry couldn't answer any of these questions. So, even though the registry informed IT about services initially deployed and made services available to consumers, it wasn't providing feedback on how these services were truly being utilized, and whether it was being done efficiently in the runtime environment.

## **A BETTER APPROACH**

Ideally, a registry ought to work hand-in-hand with complementary SOA governance offerings. This broader solution would provide the necessary “checks and balances” in the runtime environment—as the registry does in design-time. With the complementary runtime tools, IT would have the visibility to determine service reuse and performance, and have controls to ensure services were only used in the way in which the service developer and IT intended them to be used.

In RayStar's case, IT would have found—as many companies without governance are surprised to find—that their services were being used by large numbers of unauthorized consumers: so much so that application performance suffered greatly due to the amount of excessive (hidden) traffic.

---

### ***Synching Up IT and Business***

If SOA is done simply for IT's sake, not taking into account business needs, goals, rules and corporate-governance requirements, the organization will miss out on many of the benefits they might otherwise extract from the SOA. Worse—and very likely—the business will not support or be satisfied with the results.

This case study highlights is that there are two sides to SOA: technical mastery and organizational discipline. SOA projects today are still overwhelmingly initiated by IT. If that's so, how can IT hope to work with the business in order to deploy a SOA with a business-driven orientation? At minimum, the business needs to participate in the development of a roadmap of services that it can understand—and it must understand and *buy into* the benefits of those services.

And what benefits might a typical business expect to obtain? Depending on the business model, “productized” services could be deployed to make partners more efficient, for example, or value-added services could be offered externally to customers. In order to arrive at these decisions, IT must consult closely with the business—so effective processes and policies can be developed. Clearly, multilevel partnerships are essential to create a successful SOA.

### ***Getting Business Buy-In***

How to engage IT and business stakeholders in partnerships in order to develop and maintain effective projects has been the topic of much study and discussion. Top-down executive support helps. Beyond that, what are now being called Service and Integration Competency Centers (SICC)—which have the goal of institutionalizing SOA acceptance across the organization—are proving themselves to be highly effective to this end. The SICC often includes the following roles in its membership:

|                                  |                                       |
|----------------------------------|---------------------------------------|
| Business process owners          | Enterprise architects                 |
| Business analysts                | Security experts                      |
| Business domain owners           | Strategic architects                  |
| Development team representatives | Database stewards (DBAs)              |
| Product owners                   | Platform architects                   |
| IT account managers              | Professional services representatives |
| System support managers          | Line-of-business architects           |
| IT project and program managers  |                                       |

Which leads us to the next topic for discussion: Why Governance Must Be Institutionalized in the Organization or “No Thanks: We Have Our Own Standards and Processes Here.”

---

## 1. NO THANKS: WE HAVE OUR OWN STANDARDS AND PROCESSES HERE

Should governance, in the case of large, “silo-ed” organizations be parsed among separate development teams? Certainly, as this case demonstrates, there would seem to be a logic to this approach. Or should governance be “institutionalized” centrally within the organization? In fact, as we shall see, without such institutionalization, even smaller firms are taking compliance-related risks and are likely wasting resources by replicating their development efforts and especially their attempts at governance on the network.

### THE IDEA

While the “bigness” of big organizations may sometimes shield them from certain problems routinely faced by smaller firms, size brings with it its own daunting challenges—particularly when the firm is attempting to get its arms around SOA and SOA governance. In the last section, we saw how defense contractor RayStar Industries suffered from serious IT/business-related communication problems. Though RayStar is a successful and profitable enterprise, it suffers from other meaningful pains common to organizations of its size. One of those issues is how it approached establishing (or even beginning to think about) governance across its organizational silos.

### *Every Man for Himself*

To review, RayStar is an extremely diversified company: providing defense components to DoD contractors as well as aerospace products to industry. The company’s very separate operating units design and manufacture such things as components for radars, detonators, missile launchers and civil aircraft avionics.

What was not mentioned in the last section is that RayStar’s IT/business relationships are even more complicated than previously portrayed. The company doesn’t just have one overarching IT organization serving the entire enterprise. Instead, each of the several business units has its own IT team—each with its own CIO.

With this broader perspective on the IT/business “organization” at RayStar, one can almost picture the kind of organized chaos that reigned amongst the various business-unit IT teams there. Really: picture it. Each development team within each separate business unit building their versions of, say, shipping or ordering or inventory services—to suit their own product lines. Because, after all, the shipment processes around satellite components wouldn’t be remotely similar to those for detonators, right?

---

### ***King of the Hill***

And so it went. As far as the separate business-unit development teams were concerned, there was no basis for talking to the other guy. Each “local” development manager and architect was king of his own hill. And, this made sense, to a degree. One guy’s standards just didn’t seem to apply to the other. Those missile-launcher-ordering-process guys could just go ahead and keep their specialized services as far as the avionics-ordering-process guys were concerned.

### **WHY IT WASN’T SO SMART**

Even though RayStar’s development teams were working on very different services—which didn’t seem to allow them room to work together, by refusing to acknowledge each other, or even speak, they were being highly inefficient in their management processes—wasting RayStar’s resources by replicating processes—and putting the company at risk of violating regulations and policies crucial to the company’s (financial and legal) well-being by not seeking means to employ more comprehensive governance.

### **A BETTER APPROACH**

The bottom line was that RayStar—with its development teams spread out and disconnected as they were (and chose to be)—was crying out for governance. What is often unrecognized is that perhaps the most important aspect of governance is the human element. Much of what has been discussed in this paper centers around such things as “communication”, “partnerships” and “business buy-in.”

So it should come as no surprise then that the bulk of the recommendations for RayStar consist of talking ... and then setting up processes. First comes talking. This can be truly difficult—especially in organizations in which there is a great deal of political resistance associated with coming together. We might imagine this to be the case with the individual “kings” at RayStar and their hills. But there is usually a big payoff for the first two parties who finally start talking. More often than not, they are the ones who get to set the standards! (Tip: talking first can be the secret to holding onto your kingdom ...)

Once the discussion begins, it will consist over time of identifying all the existing processes, and inserting governance checkpoints and notification points (e.g., “Notify such-and-such that Service A has been created”). A standing group ought to also be incorporated to make certain that all processes are complied with. All these things said, while governance at its core is about communication, it wouldn’t hurt to deploy a SOA governance solution to simplify and automate these processes and to make absolutely certain that identified policies are complied with—particularly as these policies change over time.

---

And, most importantly, besides pairing up the IT heads from the various divisions, IT *must* pair up with business partners. To be successful over time, IT ought to choose business partners with immediate needs, who are vocal, and who can quantify the value new services can bring them. With such partnerships in hand, IT will have strong cases for moving forward in ways that will make sense to the entire organization—right up to the Board of Directors.

## **PARTING THOUGHTS**

By now, it ought to be clear that your SOA cannot do without governance. And, while the people, planning and process issues may not be simple to overcome in every instance, one can certainly learn from the mistakes of others, for example:

- > Why governance is not just a check-once, design-time concern
- > Why your organization is on the hook to comply with all relevant regulations and policies—at all times
- > Why IT teams must take an integrated approach to planning for security, management and governance policies and processes
- > How you can get around building dedicated infrastructure capacity to serve your most highly valued customers
- > How to keep track of unexpected consumers of the Web services which are now built into today's enterprise packaged applications
- > How to avoid getting lulled into complaisance by the capabilities of tools that are only part of the total governance equation
- > Why it's important for IT and business stakeholders to be in alignment from the start about the goals of the SOA and individual services
- > Why it's critical for governance to be institutionalized throughout the development cycle—and particularly across organizational "silos" in large firms.

In their efforts to provide wisdom on the subject of governance, informed sources will volunteer useful *best practices* to help the reader take the initial steps down the path toward a network of secured and managed shared services. In conjunction with such best practices, we have offered the preceding *worst practices* as a means to assist the reader to stay on course and avoid straying off the desired path.

---

***For More Information***

Learn more from Actional about SOA management and runtime governance: download more free white papers and archived webinars. Visit [www.progress.com/actional](http://www.progress.com/actional) or call (866) 438-7664 in the U.S. or +1 781-999-7100 outside the U.S. Or e-mail us at: [eval@progress.com](mailto:eval@progress.com)

MomentumSI is a Progress partner and has been at the forefront of business and technology consulting, continually setting new standards by designing and implementing integrated solutions that dramatically improve our customers' ability to drive business solutions. Visit [www.momentumsi.com](http://www.momentumsi.com) or call 888-886-8560.



