

CONTENTS INCLUDE:

- About SOA Governance
- People
- Determining your Organizational Approach
- Roles
- Policies
- Hottips and more...

SOA Governance

By Todd Biske

ABOUT SOA GOVERNANCE

SOA governance is the combination of people, policies, and processes that an organization leverages to achieve the desired behavior in their SOA adoption efforts. This refcard provides Enterprise Architects, senior IT managers, and SOA program managers with an overview of the considerations, approaches, and technologies necessary for successful SOA governance.

The hardest part of SOA adoption is the cultural change required. This can include changes not only in the technologies utilized in building solutions, but also the processes by which solutions are initially defined, funded, resourced, managed in production, and changed per the needs of the business. SOA governance must enable an organization to answer two fundamental questions. What are the right services to build? How do I build services the right way?

Beyond these fundamental questions, SOA governance must go deeper to ensure that the desired behaviors of the adoption effort are achieved. This is especially critical, because the typical desired behaviors, such as increased agility and reduction or elimination of redundancy, will not be achieved through a single project or program, but over the successful execution of many projects, and sustained through the projects that follow. It is governance that must provide the policies that guide how those projects are defined, executed, and maintained so that goals are met.

PEOPLE

Governance must begin with, but not end with, people. People must take the desired outcome of the SOA effort, determine what the desired behaviors are, and define the policies that comprise that behavior. These people must be viewed as authorities in the organization. Without authority, the organization will not respect the policies that have been established. At the same time, the selection of the people that will be associated with your governance effort and how they interact with the people executing the projects is the area of most risk in implementing SOA governance. From the people chosen, to the method of interaction with the project teams, to the decision making culture of the organization, the people can be the difference between effective and ineffective governance.

DETERMINING YOUR ORGANIZATIONAL APPROACH

When determining the people that will be part of your SOA governance effort, you must determine how those people will be organized. The challenge you will face is that an SOA

effort will touch all areas of your IT organization, and perhaps even outside of IT. It is likely that you have teams in your organization today that are involved with some aspects of governance, such as Enterprise Architecture, but it is unlikely that a single team has domain over all aspects of the SOA effort. Are your architects setting policies that determine what projects get funded? Are your managers defining policies that guide how solutions get built? The SOA effort will challenge the existing organizational structure and create dependencies across teams where they did not exist before.



The most important thing is to have your governance efforts match the culture of the organization, presuming that your organization is already effective in its decision making and policy setting processes. If it is not, then perhaps a change is in order.

Command-and-Control Organizations

Organizations that operate in a manner that always looks up the organization chart for policy definition and decision-making are typically known as command-and-control organizations. While the rigidity of the processes and recognition of the chain of command may be a turn off for many, this does not mean that the organization cannot be effective with this approach. Sometimes it is only way to get things back in line if the current processes are very chaotic and ineffective. In such an organization, having the SOA governance team show up on the organization chart either as a new entity or as a collection of dotted lines that report in to the highest levels within IT may be required to ensure that the authority of the team is recognized.



PROCESS ORDERS FASTER WITH SOFTWARE AG

Move out the right products to the right customers at the right time. With Software AG Business Infrastructure Software, you can satisfy customers with accurate, on-time deliveries. By gaining real-time visibility into where your orders are at any time, you'll see new ways to streamline your supply chain.

TO LEARN MORE, VISIT US ONLINE AT WWW.SOFTWAREAG.COM

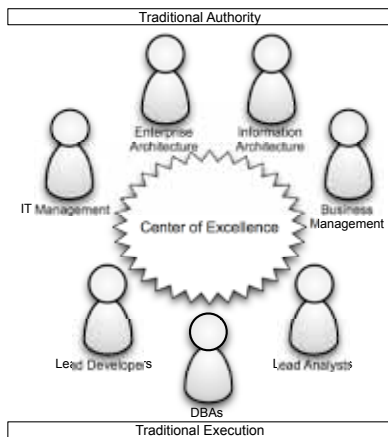
Determining your Organizational Approach, continued

Consensus-Driven Organizations

At the opposite extreme are organizations whose decision making process is completely independent from the organizational structure. These consensus-driven organizations focus on bringing the right people together for the question at hand, and allow them to reach the appropriate decision. Their effectiveness is based on how quickly the people involved can reach consensus. The risk with this approach is that decisions are not made in a timely manner, because consensus is very difficult to reach. If your organization is an effective consensus-driven organization, a formal SOA governance team may not be necessary. Rather, the right people will naturally come together to establish policy because that's how the organization operates.

SOA Center of Excellence

In all likelihood, your organization will fall somewhere in the middle. The commonality across both approaches is that the SOA governance effort must recognize that SOA will touch many, if not all areas of your IT organization, and will likely cross the existing organizational boundaries. By drawing people from multiple areas into the effort, your chances of missing some piece of the SOA governance puzzle are lessened. This collection of people will form your SOA Center of Excellence. You can choose to formally represent that within your organization chart, if necessary, or it can simply be another one of many "virtual teams" that a consensus-driven organization naturally leverages to ensure success.



ROLES

Because of the domains of policies that will be necessary for your SOA efforts, there are many roles in the organization that will need to be involved with the governance effort.

Role	Description
Enterprise Architect	Responsible for establishing technical policies, expressed in the form of reference architectures, that define how services will be built and infrastructure leveraged.
Solution Architect	Responsible for the technical leadership and decisions on a project or program, and for the project's compliance with the technical design-time policies.
Information Architect	Responsible for ensuring the consistent representation of information on service messages.
Technical Lead/ Domain Architect	Responsible for technical policies within a smaller domain, expressed through domain-specific reference architectures.
Business Analyst	Responsible for the functional aspects and business analysis activities of the solution, including defining what services get built.

Roles, continued

Security Architect	Responsible for ensuring that the technology solutions of the company take appropriate measures to protect its corporate assets and intellectual property.
Platform Manager	Responsible for the technology platform where services (and consumers, for internal consumers) are hosted, and the entry criteria for deployment.
Service Manager	Responsible for managing the relationships with service consumers via service contracts, managing the service lifecycle, and ensuring that proper run-time governance is in place.
IT Manager	Responsible for managing personnel, work, and budget.

POLICIES

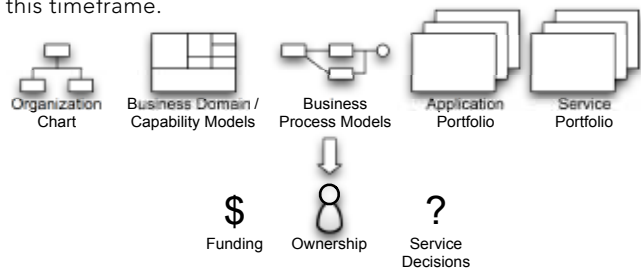
Establishing a center of excellence is not enough to ensure success with your SOA governance and SOA adoption efforts. While governance begins with people, it cannot end there. If all decisions must be made and/or reviewed by the center of excellence, your efforts will not scale as the COE will become a bottleneck for all of your projects.

In order to allow your governance efforts to scale, the people must focus first on establishing policies. Policies are the standards and guidelines that enable the staff executing your SOA projects to make appropriate decisions and achieve the desired behaviors. There are three key timeframes for which policies are necessary:

Pre-Project Governance	In this timeframe, the primary concern is with determining what IT projects to fund and execute, which is frequently associated with the broader subject of IT governance. While SOA governance should not introduce new governance processes associated with deciding what projects to fund and execute, policies associated with SOA governance should be included in the criteria. This is the timeframe where the policies must aid in answering the question, "What are the right services to build?"
Project Governance	This is also referred to as design-time governance; however, the activities in this time frame are concerned with much more than solution design. The policies at this stage must aid in answering the question, "How do I build my services the right way?"
Run-Time Governance	Governance does not end when the projects are complete. SOA adoption can increase the number of moving parts in any given solution, and if the dependencies aren't managed appropriately chaos can ensue. You need to have appropriate run-time policies to ensure the systems behave appropriately while in use and that the relationships between service consumers and service providers are managed.

Pre-Project Governance

There are key artifacts that can assist in defining policies at this timeframe.



Organization Charts	Can influence how funds are distributed in an organization, and this can be a barrier to SOA adoption. The organization chart must be leveraged and modified where necessary to account for clear service ownership and funding approaches. If the organizational issues aren't discussed at the time projects are defined and funded, it can severely hamper the efforts of the project team.
Business Process Models, Business Domain/ Capability Models and Application Portfolio	All are closely related. These are analysis artifacts that should be used to guide the decisions on what services should be created. Business process models and the application portfolio provide excellent context on existing applications and processes, but each on their own runs the risk of focusing on application silos or process silos. By combining these two with a capability map against the business domains, in essence a heat map of business capabilities, the areas where shared services will provide the most value can more easily be identified. It is important to do this outside of the context of

Pre-Project Governance, continued

Business Process Models, Business Domain/ Capability Models and Application Portfolio, continued	any project, as within a project, there is significant pressure to constrain analysis efforts and avoid "analysis paralysis." A good reference for this approach is the book "Enterprise SOA Adoption Strategies" by Steve Jones of CapGemini.
Service Portfolio	The last artifact is a catalog of services that have been built and are available in production, but it is much more powerful when it is used as a planning tool. When the organization has taken the time to perform business process analysis and business domain/capability analysis, an outcome should be the definition of key services that the organization needs to create to fully leverage SOA.

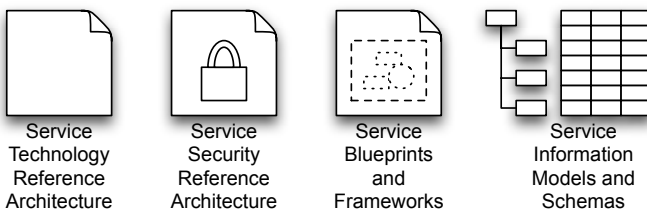
Policies for Pre-Project Governance

The following are questions/policies that you should consider in your pre-project governance efforts:

Pre-project governance
Has the proposed project identified candidate services?
Has the proposed project mapped candidate services to the business domains as represented in the business domain/capability models?
Has the proposed project reviewed the service portfolio against the list of candidate services?
Has an appropriate team of project stakeholders been identified based upon candidate services?
Has the proposed program/project been appropriately structured and scheduled to properly manage the development and integration of new and existing services?
Have all funding details been determined based upon the services proposed and the organizations involved?
Does the roadmap include the development of services with high potential for reuse?
Are projects encouraged to reuse existing services, where appropriate, based upon the business domain models and business objectives?
Are projects allowed to create redundancies, where appropriate, based upon the business domain models and business objectives?
Have existing systems been taken into account in the definition of the proposed services?
Is the organizational structure being reviewed on a regular basis based upon continued service analysis?
Does the organization have a clear approach to resolving service ownership models?
Are business processes properly leveraging services?
Does your service portfolio properly account for any globalization impact?
Does the service portfolio properly account for any planned areas for growth by acquisition?

Project Governance

In order to perform project governance, the following artifacts are recommended as sources of policy:



Service technology reference architecture

Helps ensure that appropriate technologies are used for the service being developed. It should first define the appropriate service types for the organization and then map those types to specific service technologies. Service types can include:

Composite Services	These are services that are built by combining the output of two or more services and aggregating the respective responses into a single response.
Automated (Orchestrated) Processes	These are services that are built by executing a fully automated sequence of actions as represented in a graphical process model.
Integration Services	These are services whose purpose is to enable a system that does not support service standards to speak with service consumers.
Presentation Services	These are services that provide information in a presentation-friendly format, providing information in such a way that it is easily consumed by user interface technologies.

Service Technology Reference Architecture, continued

Management Services	These are services that expose management and administrative functionality. There are technologies, such as SNMP, JMX, and WS-Management that have been tailored for this purpose.
Information Services	These are services that are used to retrieve information from a variety of data sources, aggregating the results into a single response. These services differ from composite services in that they are specifically designed to talk only to data sources on the back end, rather than any arbitrary web service.
Content Subscription Services	These are services that provide content feeds, typically adhering to feed syndication standards such as RSS and ATOM.
General Business Services	This is a catch-all category for any service that doesn't fit into any of the other categories.

These service types are mapped to service technologies, defining both the **service platform** as well as the **service communication technologies**. The service platform defines where the service implementation will be hosted and executed, such as a Java Application Server or a BPEL orchestration platform, while the service communication technology defines the message formats and the communication technologies used to interact with the service, such as XML, SOAP, and HTTP. The mapping effort links service types to technologies.

The service technology reference architecture must also provide policies on how the non-functional capabilities associated with services interactions will be provided by the underlying infrastructure. This includes:

- Security
- Routing and load balancing
- Transport and mediation
- High availability and failover
- Monitoring and management
- Versioning and transformations

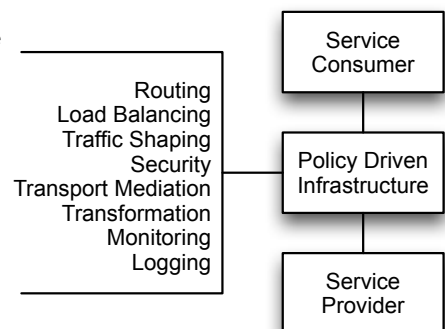
These non-functional capabilities should be enforced through policy-driven infrastructure that is configured, rather than coded.

These non-functional capabilities are a key aspect of SOA, because they are the foundation of run-time governance. At the same time, they must be factored into the design-time decisions, because if the development teams don't utilize the technology appropriately, the ability to enforce run-time governance policies will disappear.

Service security reference architecture

The next artifact is the service security reference architecture. This can be included as a subset of the Service Technology Reference Architecture, or created as a standalone artifact.

Regardless of the approach, there are two questions that must be answered by the reference architecture. What security policies must be enforced? What technologies are used for enforcing those policies?



Service Security Reference Architecture, continued

These questions must guide the developers of services and their consumers on security policies for authentication, authorization, encryption, digital signatures, and threat prevention. This includes how identity is represented on messages, how it flows through a chain of service invocations, how and where authorization decisions are made, what type of data should be encrypted and where it should be encrypted, how and where incoming and outgoing messages are checked for potentially malicious content, and more.

Service blueprints and frameworks

Service blueprints are examples of common patterns that demonstrate a policy-compliant way of solving a particular problem. Their use can make the collection of policies associated with a reference architecture much less daunting to a project team. Common patterns may include integration scenarios with legacy systems, exposing services to external parties, and consuming services provided by external parties.

Service frameworks are reusable libraries that, when used, allow implementations to be compliant with the policies of the organization, such as using the correct security credentials on service messages. Compliance is easiest when it is the path of least resistance, and making it so a developer only needs to write one or two lines of code, or even none, can make that happen.

Standard information models and schemas

Standard information models and schemas do not present one universal representation that everyone agrees on, because it probably doesn't exist. Rather, they ensure consistency in the way information is represented, minimizing the number of representations. Industry verticals, such as SWIFT (financial services), HIPAA (healthcare), and ACORD (insurance), can be leveraged as starting points, and are likely required when exposing services externally.

Policies for project governance

The following are questions/policies that you should consider in your project governance efforts, in addition to all those that were specified in the pre-project governance if not enforced at that time:

Policies for project governance
Have all services been mapped to an appropriate type?
Are the service technologies chosen for each service consistent with the type to technology mapping specified in the reference architecture?
Does the service use the standard communication technologies specified in the reference architecture?
Does the service interface comply with all naming conventions for URLs as specified in the reference architecture?
Does the service interface properly reference all external schema definitions, rather than copying them locally?
Does the service interface use the standard schema definitions properly?
Do external facing services only expose industry standard schemas, where they exist?
Is the service interface compliant with industry standards, such as WS-I?
Does the service require identity on its messages?
Are all service consumers properly specifying identity on outgoing requests?
Have appropriate authorization policies been established for the service?
Is the service communication infrastructure being leveraged appropriately?
Are all internal consumers properly leveraging the standard service frameworks?
Is all sensitive information properly encrypted according to the service security policies?

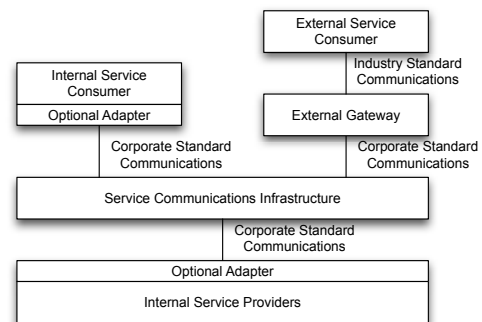
Policies for Project Governance, continued

Have service contracts been established between all consumers and providers?
Are all aspects of the service contract fully specified including message schemas, versions, delivery schedule, points of contact, and expected usage rates?
Have all services been thoroughly and adequately tested, with testing results available to service consumers, if required by the service contract? For internal consumers, testing results should always be available to help counter the natural tendency for developers to resist using things they didn't personally write.
Have service managers been assigned for all new services?
Are the service boundaries identified in the solution consistent with the business domain models?
Has the solution incorporated existing services appropriately?
Has the solution properly published information about new services into the Service Registry/Repository?
Has the solution avoided creating redundant services where not appropriate according to the business domain models?

RUN-TIME GOVERNANCE

During this timeframe, the major concern is the correct behavior of service consumers and service providers so that the infrastructure remains operational and in a healthy state at all times. At its core, the run-time infrastructure consists of three things: infrastructure used to execute the logic associated with the service consumer, infrastructure used to execute the logic associated with the service provider, and infrastructure used to allow communication between the two. Three core principles that should be adopted:

Service Consumer	Responsible for ensuring that all messages they send are compliant with the service communication standards.
Service Providers	Responsible for ensuring that they expose endpoints that can consume messages that are compliant with the service communication standards.
Service Communication Infrastructure	Enforces all non-functional capabilities for all messages that are compliant with the service communication standards, including mediation between those standards.



If a consumer or provider is not capable of being compliant with the service communication standards, adapters are leveraged not in the middle (the service communications infrastructure), but rather on the endpoints where the service consumers and service providers are deployed.

Service Contract

The **service contract** is the collection of policies that govern the interaction between a service consumer and a service provider, much as a legal contract is used to govern the relationship between two parties. This includes the messaging schemas, such as those defined in a WSDL file for a Web Service, but must also include the policies that govern the run-time behavior, such as the expected usage by the consumer in an appropriate level of detail, as well as the expected response time from the provider when the system is

Service Contract, continued

behaving as expected. It is the responsibility of the run-time infrastructure to enforce the policies of the service contract.

POLICIES FOR RUN-TIME GOVERNANCE

The following are questions/policies that you should consider in your run-time governance efforts:

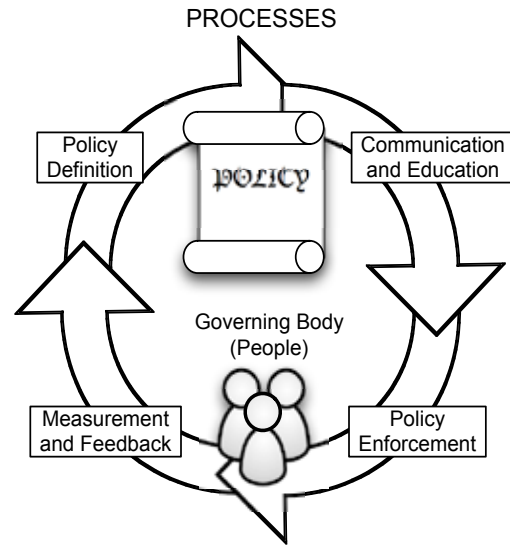
Policies for run-time governance
What is the normal rate of requests for a given service consumer?
What is the expected response time for the service provider for typical requests from that service consumer?
What actions are taken when the request rate for a given service consumer exceeds each of the agreed upon thresholds?
What actions are taken when the response time for a given service consumer exceeds each of the agreed upon thresholds?
Are there any time restrictions on when a particular consumer can access a service?
For services with multiple entry points via different technologies (e.g. SOAP/HTTP, XML/HTTP, SOAP/JMS), is policy enforcement defined and consistent (if needed) for each entry point?
Are all security policies configured and being enforced?
Are service requests routed to the appropriate version for each consumer, or have appropriate transformations applied preserving backward compatibility?
Are all service messages being logged appropriately per any enterprise auditing requirements?
Are all service messages being logged and preserved for the purpose of debugging?
Are usage metrics being properly collected?
Are usage reports being generated and distributed appropriately?
Are the recipients of these reports properly reviewing them and accounting for any discrepancies in behavior?
Are all policies associated with message structure being enforced by the run-time infrastructure?
Are non-compliant messages being logged, rejected, and reported to appropriate personnel?

THE FOUR PROCESSES OF GOVERNANCE

There are four key processes that must be executed as part of your governance processes: policy definition, communication and education, enforcement, and measurement and feedback.

Policy definition	Concerned with establishing the policies that the governance team feels will result in the desired behavior if they are followed. Without policy, the rest of the organization must either guess what the correct decisions are to get to the desired outcome, or involve someone from the governance team on every single project. The first option is unlikely to lead to success, and the second option has both scalability issues as well as being prone to variation based upon the "tribal knowledge" of the particular person from the governance team involved. Defining and documenting the policies is step one toward gaining consistency in the outcome.
Education and Communication	Just because the governance team has reached agreement and documented the policies doesn't mean they're going to be followed, or even known for that matter. A formal, planned communication effort to educate the organization on why you're adopting SOA, the desired behavior you hope to achieve, and the policies that are being put in place to achieve them is required. It's not a one time presentation to all of IT, but rather a series of targeted communications for the various roles in the organization, large group presentations, small team presentations, blogs, wikis, and appropriate surveys and followups to ensure that the communication is effective.
Enforcement	Even if your communication efforts are incredibly successful, you still need to put processes in place to ensure the policies are being followed. What you will find, however, is the better job you can do on communication and education, the easier your enforcement processes can be. If education is poor, enforcement will likely need to be more heavy-handed. Where possible, automated testing and reporting can certainly make the processes more efficient and cost-effective.
Measurement and Feedback	The governance group must have measurement and feedback processes to ensure that progress is being made toward the desired behavior. If the desired behavior is not reached, something needs to be changed, and it could easily be the policies, the processes, or the people involved with governance. Accountability is lost if the team puts policies and processes in place, but then does nothing to verify that all that effort actually paid off.

The Four Processes of Governance, continued



SOA GOVERNANCE TECHNOLOGIES

The role of SOA governance technology is not to be your governance, but rather support your governance processes by making them more efficient. To do this, there are two areas where technology can be part of your SOA governance effort.

Registry/Repository

This is the first area in support of service metadata and policy management. The registry/repository allows you to track services, service consumers, and the policies that govern their interactions. It can be the critical tool in performing service portfolio management and service lifecycle management. Through domain modeling, process analysis, and other techniques, the portfolio can be populated with planned services, updated as they are implemented, and decommissioned when all consumers disappear, with the registry/repository being the tracking point.

In addition to service metadata, the registry/repository can also be used for policy management. Policies also have a lifecycle associated with them that must be managed. If the teams that are expected to adhere to policies aren't aware the policies exist, your governance efforts will be sub-optimal.

Policy Enforcement

The second area is policy enforcement where multiple technologies can be leveraged.

Design Time

Technologies	Description
Service Testing	Can evaluate service interfaces for compliance with enterprise policies (some registry/repository solutions can also do this at time of registration). A key factor to consider is integration with other design-time testing solutions, whether for functional testing, regression testing, load testing, or performance testing. These same test cases can also be leveraged at run-time for active monitoring of services in production.
Service Framework	Shared libraries that a developer leverages when constructing service consumers or service providers. These frameworks can make policy compliance automatic, and thus the path of least resistance.

Run-Time

At run-time, there are three types of technologies that can be leveraged for enforcement of the run-time contract between a service consumer and service provider.

Service Contract, continued

Technologies	Description
Enterprise Service Bus (ESB)	enabling standards-based connectivity between consumers and providers, typically with stronger appeal to developers.
XML Appliance	These appliances tend to excel in XML security and threat protection, but may not have the flexibility of some ESB solutions.
Service Management Platforms	The focus here is on the instrumentation and analysis of service interactions, allowing alerts to be issued, requests to be throttled or prioritized, and reports generated. There is overlap with the ESB space, as many management platforms also provide gateways with similar capabilities, but the platforms typically have agents that can integrate nicely with your existing infrastructure, including ESBs, appliances, and hosting platforms.

Key Pitfalls for SOA Governance Efforts

Lack of Communication and Education	A common approach to governance is to assign some senior people to a review board and mandate that all projects go through a review as part of the development process. The problem with this is that the people involved frequently don't define the policies that teams are required to follow, and even if they do, they aren't communicated. As a result, the project teams are left to guess what the expectations are, and the reviews are likely unsuccessful.
-------------------------------------	---

Service Contract, continued

No Formal Contracts	The interaction between a consumer and a provider is more than just making sure the messages are compatible. If multiple consumers are involved, the run-time interactions must be carefully managed to ensure that one badly behaved consumer (or provider), doesn't cause all consumers to experience problems.
Lack of Service Ownership and Lifecycle Management	If the desire of the organization is to reuse existing services, nothing will stop that effort in its tracks more than having nobody responsible for the service after it is put into production. If it requires changes to support a new consumer, and no one is there to make those changes, or if funding can not be allocated properly, the project teams will take the path of least resistance and build their own version of the service.
Lack of Analysis Outside of Projects	All services are not created equal. Some services may be used by many consumers, others may only have one consumer. Treating all services as if they will have many consumers may cause an over-investment in many, just as holding off until the second consumer comes along can cause an under-investment. The proper way is to perform analysis outside of any particular project to model the organization and provide some context to make good decisions on the level investment necessary for any given service.

ABOUT THE AUTHOR



Todd Biske

Todd Biske is a Senior Enterprise Architect with Monsanto in St. Louis, Missouri. He has over 15 years of experience in Information Technology, both as a corporate practitioner and as a consultant, working with companies involved with Agriculture, Atmospheric Sciences, Financial Services, Insurance, and Travel and Leisure. His interests include Service Oriented Architecture, Systems Management Technologies, Usability, and Human-Computer Interaction. He has a M.S. degree in Computer Science from the University of Illinois at Urbana-Champaign, is a member of the SOA Consortium, is a frequent conference presenter, and writes a popular blog on strategic IT topics at <http://www.biske.com/blog/>

RECOMMENDED BOOK



SOA Governance is the key to a successful adoption of service-oriented architecture. It is the process of establishing a desired outcome for your efforts, and then leveraging people, policies, and processes to make that outcome a reality. This includes technical policies and standards that guide your design-time activities, pre-project policies that impact your project selection and funding decisions, and finally, run-time policies that impact your operational management activities. The adoption of SOA is intended to improve the efficiency and productivity of your company, and your SOA governance efforts are critical in achieving your goals in quality, consistency, predictability, change management, and interdependencies of services.

BUY NOW

books.dzone.com/books/soa-governance

Get More FREE Refcardz. Visit refcardz.com now!

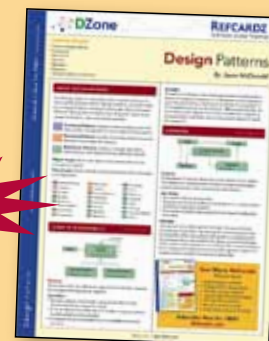
Upcoming Refcardz:

Core Seam
Core CSS: Part III
Hibernate Search
Equinox
EMF
XML
JSP Expression Language
ALM Best Practices
HTML and XHTML

Available:

Essential Ruby
Essential MySQL
JUnit and EasyMock
Getting Started with MyEclipse
Spring Annotations
Core Java
Core CSS: Part II
PHP
Getting Started with JPA
JavaServer Faces
Core CSS: Part I
Struts2
Core .NET
Very First Steps in Flex
C#
Groovy
NetBeans IDE 6.1 Java Editor
RSS and Atom
GlassFish Application Server
Silverlight 2

Visit refcardz.com for a complete listing of available Refcardz.



Design Patterns
Published June 2008



DZone communities deliver over 4 million pages each month to more than 1.7 million software developers, architects and decision makers. DZone offers something for everyone, including news, tutorials, cheatsheets, blogs, feature articles, source code and more. "DZone is a developer's dream," says PC Magazine.

DZone, Inc.
1251 NW Maynard
Cary, NC 27513
888.678.0399
919.678.0300
Refcardz Feedback Welcome
refcardz@dzone.com
Sponsorship Opportunities
sales@dzone.com

ISBN-13: 978-1-934238-44-8
ISBN-10: 1-934238-44-9



\$7.95