This is the second part of a multi-part article series. The first part is published at Driving SOA Governance - Part I.

Introduction

Many aspects of SOA governance adoption depend on tight integration between various SOA platforms. As you can see from the governance adoption framework, the SOA governance platform plays a central role in solidifying governance mechanisms, automating governance processes, and driving governance adoption. It acts as a central repository for service metadata, decision points, and current service state. It is also critical for managing design-time and run-time governance mechanisms as well as their integration with each other.

Additionally, SOA governance is concerned with how well service changes are governed, how they are propagated to production, and how much work is involved. Obviously, the more seamless and transparent are the governance processes, the less inconvenient they are deemed and, consequently, more easily adopted. Additionally, the policy enforcement mechanisms can help streamline the changes by effectively managing the SLAs for different service versions, automating service retirement, and helping transition clients to the newer service versions.

Governance Platform

Establishing the right governance platform for your organization and doing it in the right way is critical to SOA governance success. It is even more important for increasing SOA governance adoption. As discussed in the SOA governance adoption framework, the governance platform plays a central role in the whole SOA governance program.

Governance Platform Integration with an SOA Ecosystem

The governance platform should be expected to perform a variety of essential functions and deliver important benefits. These include:

- SOA governance processes automation
- integration with other SOA platform components such as:
  - cost savings
  - ROI

Quite often, the SOA governance platform also incorporates service registry and repository functions since they support the integration of SOA governance and service delivery lifecycles. Regardless of how the three aspects of the governance mechanism – governance automation, registry, and repository – are actually implemented, we will consider all of them a part of the platform in subsequent discussions.

The primary goal of the SOA governance platform is to streamline all the existing governance processes and, as a result, seamlessly weave SOA governance mechanisms into every step of the SOA lifecycle. Considering Figure 1 (from Part I in this article series), the technology target state for the SOA governance platform, let's examine how it affects each component shown on the diagram and how it drives the SOA adoption across the enterprise.
If close integration between the SOA governance platform and the rest of the SOA ecosystem can be achieved, many SOA governance processes will become an integral part of the service delivery lifecycle. More importantly, this will facilitate SOA governance adoption across the organization because the mechanisms will appear seamless for the governed population and will require little to no effort to follow.

**Governance Platform Best Practices**

Ensuring that the governance platform is deployed in the most efficient way increases the SOA governance adoption across the enterprise. Any technology that is easy to use, does not require extensive training and can seamlessly plug into the existing environment will facilitate a successful adoption of its supported capabilities.
While the same is true for the governance platform, a number of best practices should be followed to increase its viability, flexibility, and usefulness to the organization.

The adoption of SOA governance processes often hinges on the success of the SOA governance technology. If it is not completely thought through, cannot support SOA governance mechanisms, does not easily integrate with the organization's technology environment, governance processes, or SOA ecosystem, adoption will weaken. The opposite, however, is true. Effectively planning the installation, propagation, integration, and usage of the governance platform decreases the resistance and, as a result, promotes the adoption of the SOA governance processes.

<table>
<thead>
<tr>
<th>Best Practices</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>establish governance processes first</td>
<td>The governance processes should be established and executed prior to automating them in a tool. This will ensure that they work as expected and no drastic changes are needed. Manual execution of governance processes should supersede any automation efforts.</td>
</tr>
<tr>
<td>define a comprehensive metadata model</td>
<td>A comprehensive metadata model for services, policies, and governance information should be defined prior to registering a single service, defining any policies, or automating any governance processes. Without a common metadata model, there will be no consistent definition of common semantic that will be used to define services, policies, governance processes, etc.</td>
</tr>
<tr>
<td>integrate with existing governance processes</td>
<td>The governance platform should provide some capabilities to integrate with existing governance tools and processes. The tighter is the integration between SOA governance and all the related governance disciplines, the more streamlined the process will be.</td>
</tr>
<tr>
<td>integrate with the SOA ecosystem</td>
<td>As described, the target state for the SOA governance platform should provide tight integration with all the elements comprising the SOA ecosystem. These integration points should be defined upfront and integration with them should be managed through the governance platform roadmap.</td>
</tr>
<tr>
<td>concentrate on design-time governance first</td>
<td>Many organizations choose run-time governance as the first problem to tackle. This is often a mistake. Run-time governance is complex and is hard to effectively implement. Therefore, run-time governance should be explored after design-time governance practices have been solidified and automated.</td>
</tr>
<tr>
<td>create an effective governance platform roadmap and maturity plan</td>
<td>The roadmap will tell you what aspects of SOA governance to tackle first and which ones to leave for later. It will also create an effective communication mechanism to let all the stakeholders know what is happening with the platform.</td>
</tr>
</tbody>
</table>

*Table 3 - SOA governance platform best practices.*
Design-Time Governance

Design-time governance aims to address any potential service design, development, and testing issues. It primarily focuses on ensuring the following aspects of the service lifecycle.

Design

- design of the new services is compliant with the existing standards, guidelines, best practices, design patterns, etc.
- design modifications of the existing services do not negatively impact current consumers
- design promotes maximum reuse
- appropriate versioning strategies are reflected in the design

Development

- service code is in line with the existing standards and coding principles
- code complies with the approved design
- changes to the service implementation do not negatively impact current consumers
- service implementation promotes maximum reuse
- versioning is appropriately handled in the service implementation

Testing

- service’s functional and non-functional requirements are met
- service requirements are properly reflected in its implementation

The integrated SOA governance system helps achieve these goals by injecting governance automation and appropriate governance checkpoints throughout the service delivery lifecycle. Consider the expanded view of the process that incorporates all the lifecycle stages, as shown in Figure 4.
Throughout each stage in the service development lifecycle, a number of elements work together to deliver the most seamless and integrated governance experience possible. Let’s examine these elements.

1. The service delivery lifecycle is tightly integrated with the existing software development methodology.
   - as a result, integration between the two lifecycles are seamless
   - no new processes are created

2. The governance processes are fully automated.
   - the process is aware of the next governance checkpoint
   - checkpoint pass expectations are clearly laid out
   - next stage in the service lifecycle cannot start until the checkpoint has been cleared

3. The governance technology is tightly integrated into the process and increases the overall efficiency.
   - repository serves as the central service metadata storage
   - many of the SOA ecosystem components interface directly with the Repository to obtain the current state of the service
- governance automation relies on the Repository to determine next steps
- each phase in the service lifecycle adds or updates service metadata in the Repository
- policy information stored in the Repository is deployed to or queried from the Service Management platform

If you can achieve some degree of automation and integration between your SDLC, governance processes, and the SOA ecosystem, SOA adoption will increase significantly. The smoother you can make the governance process, the less intrusive it seems, and the more readily people will adopt it. This is exactly what the integrated SOA governance system delivers for design-time governance.

**Run-Time Governance**

Run-time governance focuses on the run-time aspects of the service. This primarily involves ensuring compliance with the established policies, managing SLAs, and proactively monitoring service performance.

Organizations often start with the run-time governance as the first component in their SOA technology stack. This is because it provides the most visible and easily understood value proposition – service management. Regardless of how services are developed, governed, and deployed, managing their run-time execution stabilizes their operational aspects, increases scalability, and improves monitoring. Thus, any time organizations where run-time governance technology is deployed find that SOA adoption is amplified due to its highly visible benefits.

The success of run-time governance hinges on several factors.

**Platform Deployment**

- ability to handle the combined load of all the services under management is key
- flexibility to easily adopt to SOA ecosystem changes needs to be built-in upfront
- product should be highly scalable and provide high performance
- easy integration with existing or future Registry, Repository, and ESB products should be a feature of the platform
- ability to automatically govern service execution is a bug plus
- ability to capture run-time metrics can add significant value

**Policy Management**

- ability to manage service policies through the tool itself or obtain them from a Repository is critical
- proactive policy enforcement adds a lot of value
- propagation of policies throughout all the nodes helps increase performance and scalability
- compliance with policy and security standards increases adoption

The ultimate goal of achieving the integrated SOA governance should be kept in mind when implementing run-time governance. The governance mechanism will supply the service management platform with all the necessary metadata and ensure that no service can be accessed if it does not have the right permissions or has not completed all the governance steps. The Repository will become the central policy store and will feed all the approved policies to the service management platform. It, in turn, will supply the Repository with all the run-time metrics.
Since the value of run-time governance is easily recognizable, the simple fact of its deployment increases SOA governance adoption. However, adoption can be further increased if all the design-time governance processes are made a prerequisite for run-time governance enforcement. This closes the loop on not only building the services correctly but also ensuring that they are executing as expected in the production environment. With the integrated SOA governance system, no additional steps are necessary to enact run-time governance. It simply becomes an extension of design-time governance processes. As a result, the entire SOA governance lifecycle becomes transparent and easily adopted.

Example

To-date, the McPherson Corporation has implemented a lot of governance mechanisms and has been fairly successful in shoring up the adoption in its own IT department and those of its subsidiaries. McPherson IT has implemented most of the SOA platforms on its roadmap that included the ESB and Registry/Repository with integrated SOA governance automation capabilities including both design-time and run-time governance. No independent service management platform has been selected as the Registry/Repository run-time governance is slated to manage the run-time policies and the monitoring tools are expected to alert about any noncompliance situations.

Now, that SOA governance has become a reality and is well received across the enterprise, the SGPO decides to take advantage of the SOA governance automation capabilities that have remained dormant to date. The first step it takes to achieve this is to create a comprehensive metadata model that would capture all the service elements throughout the entire lifecycle. An initial model is created by the members of SGPO, Enterprise Architects, Service Architects, and Service Analysts. It is then ratified with all the stakeholders and groups affected by the automated SOA governance process through a series of meetings and discussions. The complete metadata model is implemented in the Repository and service delivery governance processes are automated.

As the first attempt to automate SOA governance processes, SGPO decides to implement the following automated governance checkpoints:

• approve candidate service checkpoint
• validate service design checkpoint
• validate service deployment checkpoint

While all the attempts have been made to integrate the ESB with the governance automation states, little progress has been made on this front. The automation allows the service to be successfully deployed to the production environment and exposed through the ESB but each invocation does not check for proper status flag in the Repository.

This problem is minimized, however, through the use of run-time governance aspects of the Registry/Repository platform. The automated governance processes mandate that service policies are developed. Once the deployment to production takes place, all of the run-time policies are pushed to the run-time management platform and are enabled as a result.

While the overall process is not 100% perfect, the SGPO realizes that it is only the first step in a long road towards full maturity. Before the integrated process is tested, the SGPO and the Service Registry Custodian register all the existing services and their metadata attributes with the Registry. The Policy Custodian performs the same work for all the existing policies. Now, the platform is ready for customers.

SGPO selects the Tri-Fold SOA team to be the first to test out the new integrated process. Several new services and enhancements are identified as a pilot group of projects. SGPO, the custodian group, and enterprise architects all work closely with the Tri-Fold SOA team to ensure the pilot is executed smoothly and
its findings are immediately put into practice.

As a result, several key issues are uncovered.

- More comprehensive Service Certification Review is needed.
- Deployments to non-production environments should include policies and registration of services with the corresponding platforms.
- Service versioning and retirement mechanisms should be enhanced.
- Specific governance mechanisms and policies should be created around service orchestration.
- There is a need to provide a formal appeals process for governance decisions.

The processes are modified to address these issues and the integrated SOA governance system is rolled out to the entire enterprise. Within the first three months, teams within McPherson IT have reported a greater degree of compliance with SOA governance processes, increased efficiency, and faster delivery cycles. The metrics being collected by the SGPO have supported these findings. Within the first six months, similar results are observed in the Tri-Fold IT environment. SGPO credits this fast success rate to the initial pilot that was done with Tri-Fold SOA team. SGPO’s goal is to continue these trends in its other subsidiaries including Alleywood where political realities have prevented strong governance to be exercised in the past.

Conclusion

The SOA governance platform should seamlessly integrate with the entire SOA ecosystem and SOA governance processes should be established before any technology is selected or implemented. Automating and integrating SDLC, governance processes, and the SOA ecosystem, will increase SOA adoption, especially if all the design-time governance processes serve as prerequisites for run-time governance enforcement.

The next article in this series will address organizational aspects of SOA governance.

About the Author: Leo Shuster

Leo Shuster is responsible for SOA strategy and execution at National City Bank.

He has over 15 years of IT experience and throughout his career, has performed a variety of roles including group manager, team lead, project manager, architect, and developer.

His experience spans a number of industries but over the past 7 years, he has been focused on the financial services sector.

Mr. Shuster holds a Masters degree in Computer Science and Engineering from Case Western Reserve University and an MBA from Cleveland State University.

He has presented on SOA and related topics for groups of all sizes at such events as the Gartner summits and the Enterprise Architecture Executive Council.

He regularly blogs about advanced software architecture issues at leoshuster.blogspot.com and he can be reached at leo.shuster@nationalcity.com

www.servicetechmag.com/contributors/leoshuster