

# Five Phases to Support the Data-Centric SOA

By Adel Harris and Ramesh Ramakrishnan

**W**e've all heard that the successes of enterprise initiatives are only as good as their data foundation – and the transition to a Service Oriented Architecture (SOA) is no exception. As organizations look for strategies to manage data as enterprise assets, an evolutionary, data-centric SOA transition allows them to move one step closer to this goal.

Delivering on the promise of true enterprise-wide SOA – an environment in which discoverable, accessible, interoperable, and trusted services are business applications that are aligned with business goals – is a huge undertaking that takes years. But with a proven, evolutionary SOA transition plan that focuses *first* on data, organizations can achieve sustainable results with “quick wins,” while proceeding down the path of an architected approach that will deliver fully on SOA's promise.

The formula for SOA success is not a one-size-fits-all approach and it's not merely a technical, system-centric solution for application consolidation. At the heart of your SOA transition must be practical strategies that begin with creating a core of minimal architecture requirements –with data as the foundation.

A data-centric SOA transition allows organizations to better leverage new and existing IT investments to support business requirements, transition legacy systems incrementally, and deliver reusable, interoperable, secure, trusted business services.

This article outlines the five iterative phases of a data-centric SOA transition that has become a best practice for several organizations.

SOA enables new and existing enterprise systems to share services and information across technical platforms, departments, and ultimately across organizational boundaries. To achieve the benefits of SOA, organizations must transition from a siloed system-centric view to an enterprise data-centric view of IT. This transition requires that enterprises break down the barriers between business and technical

organizations that have prevented sharing information and IT solutions in the past.

## SOA Formula: Focusing on Data

A data-centric SOA transition looks at SOA from more than just a technology perspective and builds a SOA strategy around authoritative data sources.

By focusing on the data that is at the core of legacy infrastructure, a data-centric strategy preserves your IT investment and provides better access to legacy systems by integrating existing systems and tapping into authoritative data sources. Extending the existing application by adding new functionality allows for an agile implementation that is responsive to your business objectives. And migrating your business functions/processes to the new platform preserves business functionality and reduces support costs by moving to an enterprise technology solution.

Designed to mitigate challenges enterprises face when implementing SOA, a data-centric methodology combines legacy modernization efforts, metadata management, and data-quality best practices to help organizations infuse an authoritative data sources approach into their programs.

This approach is based on the dual premises that services must align with business and information requirements, and that they must be discoverable, accessible, interoperable, and trusted in order to deliver on the promise of SOA.

A data-centric approach consists of five iterative phases to support the evolutionary transition to a SOA environment that are held together by a robust governance process. This five-phased approach mitigates the risks of implementing an SOA and provides a roadmap to design, acquire, orchestrate and govern services that fit the needs of your business modernization and integration efforts.

These phases include: Architect and Plan, Decouple Data, Model, Assemble, and Deploy and Manage.

## Phase I: Architect and Plan

Services, new or acquired, need to be aligned to an organization's target business architecture so that enterprises can facilitate information sharing by breaking down the barriers between business and technical organizations. Establishing an architecture that provides the framework for business and information requirements, and integrates processes and data from literally hundreds of systems needs to be done to effectively implement SOA.

The Architect and Plan phase focuses on architectural alignment, gap analysis and transition planning. Architectural alignment identifies how the scope of the SOA efforts fit into the target architecture, and categorizes current systems and data stores into that framework to identify opportunities for reuse and consolidation. The architectural alignment is also used for gap analysis, identifying high-level requirements to build or acquire services, and planning the transition from the current to the target SOA environment.

During this phase, the organization creates a transition roadmap that will provide an enterprise view of business objectives and insight into where data is being entered and used again and again. (cont.)

---

*Adel Harris is the enterprise architecture solutions director at Citizant. Harris is a certified federal enterprise architect and a recognized thought leader in practical information-sharing solutions and EA best-practices that connect the dots between business mission and IT solutions.*

*Ramesh Ramakrishnan is a division director at Citizant, where he oversees the development and delivery of customer solutions for the U.S. Departments of Housing and Urban Development (HUD) and Transportation (DOT) and other government agencies.*



## Phase II: Decouple data

The next phase focuses on decoupling data from systems to provide the basis for an information warehouse and support an incremental transition. This step depends on the requirements of your transition and your enterprise environment. In environments where multiple disparate systems are not interdependent, it makes sense to skip the decouple phase and move to Phase IV.

However, decoupling data is important in cases where many enterprise systems rely on the same set of data. Decoupling the data breaks this point-to-point contact and makes your enterprise systems more data dependent. Data becomes a shared enterprise resource that is stored at a centralized place and not dependent on any one system. This provides the basis for creating authoritative data sources and gradually evolving to SOA. This involves

- classifying data into existing high-level information classes
- identifying systems/functions affected based on business drivers
- publishing decoupled data with information services of affected systems/functions and
- implementing business rules to decouple the systems.

By focusing first on the data definitions and relationships, you can ensure that subsequently designed services originate from authoritative sources and that data exchanged are present in a centralized/integrated “data store,” which makes subsequent data modeling easier.

## Phase III: Model

The Model phase focuses on adding more depth to the target architecture and creating the core requirements for the service. This entails evaluating current legacy assets in terms of how they are supporting business utility then categorizing investments, defining duplication, formulating criteria for selecting best-of-breed and ultimately laying the foundation for a true enterprise-wide framework.

Working from the core requirements, this approach ensures that consolidation makes sense and that unique business unit processes and requirements are maintained. It provides the basis for evolving requirements to meet overall enterprise objectives.

Modeling involves harmonizing existing systems and data sources into the context of the architecture to establish standard business rules, data structures and semantics for the enterprise. Model diagrams are the basis for building, generating, or inte-

grating software for the data sources, services and communications.

## Phase IV: Assemble

The Assemble phase focuses on using the detailed requirements for the services and their interfaces documented during the Model phase to assess the “fit” of current IT assets to identify solutions that should be reused, harvested from existing systems, created or acquired for the target SOA solution. Thinking through the detailed requirements helps to define the assembly and orchestration of services to support target business processes, and the assembly of services to build applications.

## Phase V: Deploy and Manage

The organization leverages proven best practices and industry standards to determine migration patterns to keep systems as is, but with a newer front-end and new functionality as evolutionary changes or extensions.

Planning deployment includes registering metadata for service consumers, establishing policies and measures for service contracts with consumers, and supporting the maintenance and release management processes. The actual deployment and monitoring of services in a SOA environment is typically handled by a centralized operations group.

## Robust Governance Process and Quick Wins

These iterative transition phases are held together by a robust governance process. Most organizations will agree that managing IT from an enterprise perspective coupled with an inventory of services in constant motion makes governance more challenging and integral to the success of SOA. New tools, standards, management processes, and life cycles need to be established and governed to ensure the sustained and effective use of SOA across the agency.

With a data-centric approach, governing an SOA implementation is conducted at each phase by establishing milestones for compliance reviews. Before governance can be executed, it must be defined, approved and communicated. Effective governance requires participation from subject matter experts to define components such as standards for services, processes for monitoring and managing service performance, methods to assure information quality and security, and policies for change and release management. Governance also requires executive sponsorship, investment

control, business stakeholders, and inter-constituency working groups to institutionalize and ensure the effectiveness of SOA.

Additionally, “quick wins” are important to sustain the momentum and support for SOA. Examples of

potential “quick win” opportunities include sharing and re-use of common service components, consolidation of redundant applications and initiatives and refresh or replacement of non-standard technologies. Using an evolutionary transition to SOA allows organizations to achieve “quick wins” that deliver sustainable and measurable results to the agency, while at the same time proceeding down the path of an architected approach that will deliver fully on the promise of SOA.

## Conclusion

Despite the challenges associated with planning and deployment, SOA has the potential to deliver great value to organizations. The ultimate destination of SOA is achieving an environment in which discoverable, accessible, interoperable, and trusted services represent business applications that are aligned with business goals. The roadmap for this journey is charted through a phased, evolutionary data-centric transition.

SOA is a planned and architected approach that provides agile IT solutions aligned with programmatic requirements. A data-centric approach to achieve and sustain the promise of SOA is based on understanding the business and information requirements and aligning services – whether harvested (from legacy applications), acquired (COTS/GOTS), or built from scratch – based on these business drivers. This approach must be supported by architecture, planning, business models, standards and governance to deliver reusable, interoperable, secure and trusted business services to internal and external consumers.

SOA offers enormous potential for organizational agility and information quality. Getting there requires organizations to develop practical data-centric principles and processes that will enable them to connect the dots between business mission and IT solutions. These principles and processes provide the framework for integrating existing systems, re-using existing functionality as services, and leveraging authoritative data sources to improve efficiently, quality, and manage costs.