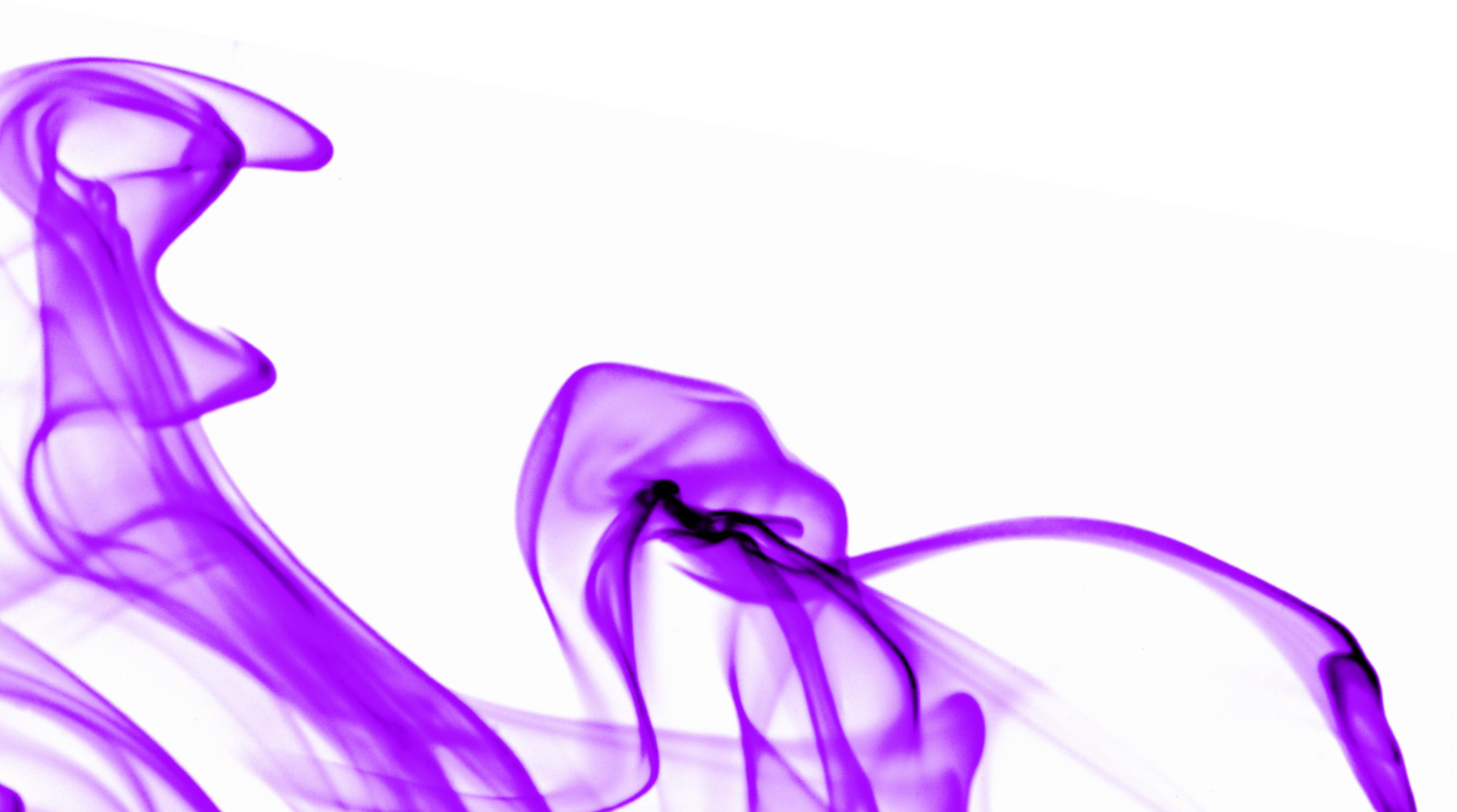
BEA White Paper

Deliver Immediate Value Through Service-Oriented Architecture

Service-Oriented Architecture is as Much a Business Innovation as it is a Technological Innovation



Copyright

Restricted Rights Legend

Trademarks and Service Marks

CW1432E0906-1A

Contents

Introduction

Service-Oriented Architecture is as much a business innovation as it is a technological innovation. It has the potential to erase the barriers that separate the business and technology sides of any organization, dramatically improving productivity, and increasing shareholder value. This can be accomplished with a relatively modest, incremental investment in information technology.

How does SOA increase shareholder value? In a word: agility. A well-managed SOA enables rapid response to changing market conditions.

SOA is a design discipline that facilitates componentized application development. Applications composed of discrete Web services simplify the modification of business processes and speed the delivery of new composite software solutions in response to marketplace conditions and competitive challenges. That combination of speed and simplification adds up to an improved bottom line.

But there is a more important element: when managed as a strategic architectural initiative, services can reduce the complexity of existing software systems. That complexity cripples business agility and can add millions of dollars to the cost of system maintenance, which already consumes as much as 80 percent of the typical IT budget. The IT productivity gained through a properly managed SOA can be channeled toward strategic goals and future business solutions, rather than bandaging past technical problems.

But the key to realizing the value in SOA is management. The transition to SOA takes time and requires the proper planning. SOA management extends beyond the availability and performance of individual services. Effective SOA management also requires:

- Managing services as part of the overall organizational ecosystem
- Promoting SOA as the future state
- Governing Web services and service usage from design time through runtime
- Measuring and communicating value

In an increasingly unforgiving global marketplace, no organization can afford the luxury of waiting years to derive value from SOA. Fortunately with the right tools, it is possible to work toward strategic goals and the service-oriented vision while delivering measurable value today. This paper presents four practices for managing SOA in order to deliver both immediate value and sustained return on investment.

Practice #1

Manage SOA as part of an ecosystem

SOA provides a layer of simplicity over a system of complexity. But SOA does not stand alone, nor can it be managed that way. If treated in isolation, Web services will only exacerbate organizational complexity. Any attempt to manage Web services in isolation is a grave mistake.

Succeeding with a Service-Oriented Architecture requires an understanding of the complexity surrounding it—a view of SOA interconnected with the organizational ecosystem:

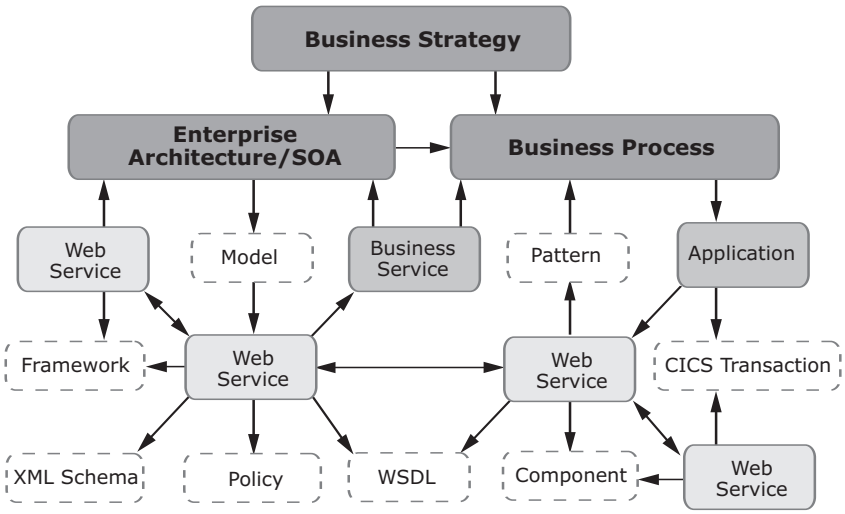
- SOA supports the business architecture
- SOA and underlying services orchestrate business processes
- SOA is part of the technical architecture
- Applications and systems interact with the SOA and underlying services
- SOA and services are composed of components
- Internal teams, external customers, and trading partners access and reuse services and the SOA

The role of the SOA repository

In order to understand how services fit into the larger ecosystem, it is necessary to identify and document the relationships between services, the business processes they orchestrate, the systems and applications they interact with, and their constituent components. This task requires an enterprise SOA repository, one that will articulate the relationships between services and other assets, and provides a dynamic, graphical view of these relationships.

By conveying an understanding of the role that services play in the organizational ecosystem, this practice represents an important step in reducing system complexity. It allows stakeholders to identify interdependencies within the ecosystem, which facilitates informed decision-making and impact analysis.

SOA and Web services are interconnected with a multitude of assets in the IT ecosystem.



Manage SOA as part of the overall enterprise architecture

Enterprise architecture links the business structure to the technology infrastructure, which includes SOA. Developing the SOA independent of the enterprise architecture is like laying out plumbing and electrical systems for a building that doesn't exist. Without a building plan, how will the contractors know where to put the pipes and wires? Managing SOA as part of the EA facilitates the identification of the business functions that will be best served by the SOA. It is then possible to prioritize and build out the SOA based on the business functions that are part of the organizational strategy. This allows clear justification of SOA investments, and lays the groundwork for the measurement of the returns from SOA, in terms that are meaningful to the business.

For example, assume that the business strategy is to grow through mergers and acquisitions, and to begin presenting a common face to the customers as soon as possible. All customer-facing business functions will be affected by any acquisition. The SOA might support this strategy by providing an infrastructure to quickly allow the sharing of customer data between the parent organization and the new acquisition, and provide a common interface to all customers. The value of the SOA might be measured in terms of cost savings and time savings associated with the technical side of the acquisition, and the incremental revenue that the organization realizes by integrating the acquired systems in six months rather than two years.

Enable reuse

Reuse extracts the maximum value from existing services, while at the same time working toward the overall SOA.

Internal developers, external customers, and trading partners need to reuse services when designing systems and functionality. To enable reuse, services must be published in the repository, with usage histories, test results, peer feedback, service performance metrics, and other meaningful metadata and supporting documentation. Web service performance metrics should be dynamic, allowing stakeholders to view both design time and runtime information about the service.

Service consumers should be tracked through the repository. This will provide a good feel for the required service level, facilitate impact analysis, and allow stakeholders to be notified proactively of any service changes.

Publishing upcoming services in the repository will facilitate collaboration and allow teams to plan for reuse. The repository should automatically notify existing users and interested parties of new service releases.

Practice #2

Promote SOA as the future state

Realistically, SOA is never done. The target architecture must constantly evolve in order to accommodate changes in the external business environment and the corresponding adjustments to organizational goals. Similarly, the underlying services should also be continually reprioritized to adapt to the changing business environment. Value is derived from transforming from the existing architecture to the target architecture. The first step in managing this transformation is the communication and promotion of the target architecture.

The target architecture is the IT roadmap. It provides a dynamic view of the future state of the organization in real time. Publishing and maintaining the target architecture in the repository provides stakeholders with a frame of reference to determine where they are today and where they need to go tomorrow.

The target architecture is comprised of capabilities that will be fulfilled by services. The target architecture and the underlying capabilities should be published in the repository with the services. The repository provides stakeholders with easy access to view the target architecture and understand the included capabilities, and to view existing and upcoming services.

As an example, imagine once again a company that wants to grow through mergers and acquisitions. This strategy requires the company to share customer data and to present a consistent face to its customers. The target architecture calls for an interaction platform that will provide a set of common services to facilitate customer interaction. These services will extend across multiple channels of customer interaction, including Web-based, voice response/telephone, store contact, and kiosks. The published target architecture will illustrate the interaction platform, and will dynamically link to all current and future services to be included in the platform. The services must be described with metadata. One example of the required metadata fields is the service status, which will indicate whether the service is under development, available, or targeted for future development. A target availability date will be provided for all services either under development or targeted for future development.

Publishing the target architecture allows stakeholders to:

- Identify and reuse existing, architecturally compliant services
- Assess the risk associated with using upcoming services
- Monitor the progress of services that are under development
- Determine the impact of upcoming changes

Practice #3

Govern services and service usage

Publishing the target architecture in the repository allows development teams to identify where they are today, and where they need to be tomorrow. However, these teams won't necessarily know the best route to achieve architectural compliance. Architects are well positioned to identify and publish standards, practices, and policies, and to craft architectural blueprints that serve as a roadmap for development teams. These architectural blueprints should be included in the repository.

For example, in order to present a common Web-based face to customers, the general solution might involve using a common presentation layer and the Web-based interaction service. This solution will be prescribed, or specified, for any project that will build a customer-facing Web site. Prescribed solutions for specific projects will be provided to the project teams via the repository. Usage tracking data collected by the repository can then be used to determine the level of architectural compliance.

This approach provides architects with a non-invasive means of monitoring and tracking compliance—allowing them to focus on evolving the SOA rather than policing development.

The same approach can also be used to support the tasks performed by architecture review boards.

Monitor exceptions

It is impossible to achieve full architectural compliance across the entire enterprise. Situations will arise in which architectural standards cannot be implemented and the prescribed solution cannot be deployed. However, if these instances are tracked, they can be addressed in the future.

For example, if a development team building customer-facing Web sites discovers that it can not use the common presentation layer, the architecture team must be informed. Often, non-compliance is identified through architecture review boards and tracked manually. However, if the architecture has been published, solution sets have been prescribed to projects, and usage of those prescribed assets has been tracked, compliance tracking becomes an automated function. Teams that cannot use the prescribed solution can be easily identified and issued compliance waivers.

Formalizing and recording architecture waivers legitimizes the architecture process with the recognition that project priorities can outweigh architectural compliance in the short term. Flagging and tracking out-of-compliance IT investments also enables continuous review and presents the opportunity to enforce long term compliance of those projects.

Enforce policies

Any discussion of the governance of service usage must also address the governance of services themselves. Services must comply with organizational policies and meet the performance criteria specified in service level agreements at design time and runtime. Enforcing policy and compliance at design time can avoid an interruption in service at runtime.

For example, an organization might have a performance policy for production services that specifies the minimum requirements for production Web services. Measurements relating to this policy might include maximum users, total throughput, average hits/second, and average response time. These requirements must be considered at design time in order for the service to perform at runtime. In addition, actual runtime service performance levels must be compared to the performance levels specified in organizational policies in order to identify services that consistently fall below the minimum requirements.

Practice #4

Measure Progress and communicate value

All too often, the CIO will communicate a service-oriented vision of tomorrow, and architects will hole up for a few years working toward the CIO's vision. However, as previously noted, the target architecture is never complete. Rather, it is constantly evolving—as it should be.

But the CIO needs to know how much progress has been made toward fulfilling the current vision and how much funding is required to complete it (provided the architecture remains stable). The CIO also needs to know how much value has been derived from the architectural investments to date.

Measure progress and determine future funding

As stated in Practice #2, the status of services, whether planned, funded, or complete, can be aligned with the target architecture. This provides a means of reporting progress. Reports generated by the repository should summarize the current status of capabilities against the plan. This provides architects with the means to quantify and articulate their progress toward realizing the living target architecture, and identify the capabilities and services that still require funding.

The value derived from the architectural investments

Value from an effective SOA is delivered across various dimensions. The value derived from reusing services can be expressed in terms of development productivity savings and as ongoing maintenance savings. Development productivity can be translated into improved quality, faster time to market, or increased functionality. This provides the means to quantify the value derived from the reuse of existing services.

Value is also derived from the adoption of standards. Standards ensure interoperability and consistency. The adoption of standards can be measured as the percentage of projects that are compliant with the target architecture. Standardization also creates value through the reduction of overall system complexity.

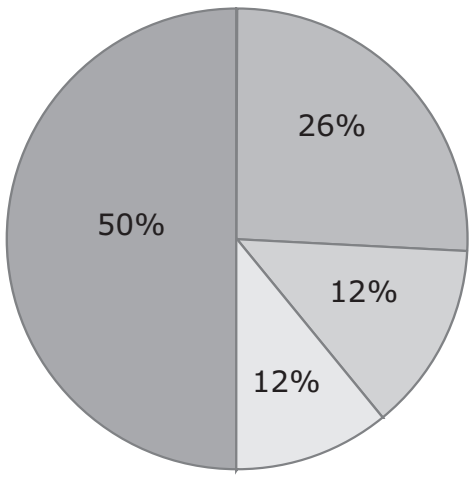
Support investment decisions by measuring progress toward the target architecture.

50% Complete
Investment = $476,000

26% Under Development
Est. Investment = $247,520

12% Planned
Investment = $228,480

12% Proposed
Est. Investment Unknown



The greatest value of SOA is also the most difficult to directly measure. It is derived from the support that the SOA lends to the business strategy—as determined by the speed with which the integration of an acquisition can be completed with the SOA—versus without—and the impact of this productivity improvement on shareholder value. Collection of the necessary information requires alignment and integration between the project portfolio (project portfolio management tools) and the asset portfolio (enterprise repositories), creating a chain of evidence that proves that the envisioned value is actually being realized.

Conclusion

Organizations that successfully implement SOA will emerge in a form far better equipped to thrive in the fluid and unpredictable 21st century business environment. Ultimately, the degree to which SOA can deliver its full potential will be determined by the implementation of the practices outlined in this document:

- Manage services as part of the overall organizational ecosystem
- Promote SOA as the future state
- Govern services and their usage from design time through runtime
- Measure progress and communicate value

The key to successfully implementing and maintaining these practices is an enterprise repository capable of organizing, managing, and measuring the use and reuse of Web services, SOA, and the full spectrum of software assets.

About BEA

BEA Systems, Inc. (NASDAQ: BEAS) is a world leader in enterprise infrastructure software, delivering unified SOA platforms for business transformation and optimization. Customers depend on BEA Tuxedo®, WebLogic®, and AquaLogic™ product lines to help reduce IT complexity and leverage existing resources—for achieving a state of Business LiquidITy™ where enterprise assets are freed up to deliver maximum business value and grow new revenue streams. Find out more at *bea.com*.

Join the BEA community

At BEA, we understand that developers need different kinds of resources than IT managers. And that architects face different challenges than executives. That's why we've created four unique communities that give you exclusive access to a formidable group of your peers, to a world of shared thinking, and to the kind of meaningful information that can make you more effective and more competitive. To join one or more of the BEA communities, simply register online at *bea.com/register*.

BEA Systems, Inc.

2315 North First Street
San Jose, CA 95131
+1.800.817.4232
+1.408.570.8000

bea.com

CW1432E0906-1A