Organizational Adoption of SOA Best Practices

Ron Schmelzer & Jason Bloomberg
Senior Analysts
ZapThink, LLC

Take Credit Code: DKORG

The Problems of IT are The Problems of Business

Are YOU doing all you can?
Business Constant: Change

Changing Marketplace  
Competition  
Mergers & Acquisitions  
Business Partners

Customer Demands  
Changing  
Optimizing Processes

New Technologies

*A Business is Never STATIC*

We’ve had IT challenges for years ...
... but even after yesterday’s promises...

... we still have the same IT mess, only worse.
The Business Inflexibility Trap

- **Inflexibility** is the Mother of All Business Problems
  - If you’re flexible enough, you can solve all the other problems

- Information Technology (IT) is an impediment to business change
  - It wasn’t supposed to be that way!

Business Agility

- Companies require *Business Agility*...
  - Responding quickly to change,
    and
  - Leveraging change for competitive advantage

*Agility is the key to innovation*
What do you Want your IT to Do Anyway?

- The Automation Paradox
  - The more we automate, the more our remaining problems are difficult to automate

- Is IT about getting the technology to work together or to help the business meet its goals?

**Who’s in control of IT, anyway?**

Service Orientation:
Light at the End of the Tunnel

- Service Orientation is a *business* approach
- It’s not about connecting things, it’s about enabling processes
- The core business motivation is business agility
- Rather than “rip and replace” old systems – make them work better together
- We don’t want more middleware for our middleware
Level Set – What is SOA?

- SOA is architecture – a set of best practices for the organization and use of IT
- Abstracts software functionality as loosely-coupled, business-oriented Services
- Services can be composed into business processes (which are also Services) in a declarative manner
- As fundamental a change as mainframe to client/server or client/server to the Internet

What is a “Service”?

- Broad meaning: something providing a capability to another as needed
  - IT Services, like print services or email services
  - Business Services, like accounting or human resource management
  - Software-as-a-Service
- Narrower meaning: an abstraction of IT functionality or data
  - Web Services are one example
  - SOA starts with this kind of Service
The Benefits of SOA...

- Reduced cost of integration
- Improved value from legacy applications
- Reuse leading to reduced redundancy
- Greater visibility for governance & compliance
- Increased reuse of software assets
- Business agility

Key Areas of SOA Investment

- Reduction in integration expense
  - EAI replacement/EAI maintenance reduction
  - Legacy enablement/migration/rejuvenation
- Increase in reuse
  - Reduction in redundancy
  - Better customer visibility
- Increase in business agility
  - Improved competitiveness
  - Faster innovation
- Enablement of governance & compliance
Have We Been Here Before?

- Service-Oriented Architectures have been around a while
- CORBA (Common Object Request Broker Architecture) and DCOM (Microsoft Distributed Component Object Model) two familiar examples
- What’s new this time?

The Difference is Web Services

- WS-Policy
- WS-MessageData
- WS-Transfer
- Web Services for Remote Portlets
- WSDL
- Web Services Metadata Exchange
- WS-Addressing
- BPEL and BPELJ
- Web Services Business Activity Framework
- WS-Coordination
- Web Services Eventing
- WS-AtomicTransaction
- Web Services Coordination
- Web Services Transaction
- WS-BusinessActivity
- WS-Choreography Interface
- SOAP
- WS-Trust
- WS-Federation
- WS-SecureConversation
- WS-Acknowledgement
- WS-ReliableMessaging
- WS-ReliableMessaging
Web Services are the Trees....

Service Orientation is the Forest

How to Think Service-Oriented

- Service-Orientation is about change
- IT must respond to change and enable innovation

Rather than simply throwing more software & iron at the problem, we need a better way of organizing IT resources
Business Process the Old Way...

- People plugged into rigid processes
- Inflexible & brittle

Business Process the Service-Oriented Way...

- IT resources (among other resources) available to the business as needed
- Business users create composite applications by composing Services on the fly
SOA Abstracts the Plumbing

- The goal is reusable, composable business Services
- Many different approaches to implementation

SOA shifts the way we think

<table>
<thead>
<tr>
<th>Traditional Distributed Computing</th>
<th>Service Oriented Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designed to last</td>
<td>Designed to change</td>
</tr>
<tr>
<td>Tightly Coupled</td>
<td>Loosely Coupled, Agile and Adaptive</td>
</tr>
<tr>
<td>Integrate Silos</td>
<td>Compose Services</td>
</tr>
<tr>
<td>Code Oriented</td>
<td>Metadata Oriented</td>
</tr>
<tr>
<td>Long development cycle</td>
<td>Interactive and iterative development</td>
</tr>
<tr>
<td>Middleware makes it work</td>
<td>Architecture makes it work</td>
</tr>
<tr>
<td>Favors Homogeneous Technology</td>
<td>Leverages Heterogeneous Technology</td>
</tr>
</tbody>
</table>
SOA’s Little Secret

- You can build SOA with whatever software you already have (for the most part)
- When considering purchasing software for your SOA initiative, look to fill the functionality gaps in your current portfolio
- Don’t just add middleware for your middleware!

Don’t Get Lost in the Terminology

- There are many styles for SOA implementation
- Focus on your goals: Reuse? Governance? Reduced integration cost? Agility?
How Do You Eat an Elephant?

- *One bite at a time!*
- Don’t expect to have all the answers on day one
- Take a step-by-step approach
- Show business value at each step

Building SOA the Right Way: Take an Iterative Approach

- Top-down only: have the plan, may not be able to execute
- Bottom-up only: build Services, may not be reusable
- SOA planning *must* be both
  - Develop the vision (but not the details) ahead of time
  - Service development should be iterative
The ZapThink SOA Roadmap

- One way to help lower your risk
- Steps should overlap
- Your roadmap will vary

Challenge: SOA is Architecture

- Remember...SOA is architecture – in particular, *Enterprise Architecture*, including:
  - An aggregated architecture of all the individual IT systems within an organization
  - The human element within the enterprise
  - Systems, people, and organizational constructs at other companies that have relationships with the enterprise
  - Individual consumers who are that enterprise’s customers
  - Corporate governance
Is there an Architect in the House?

- The new discipline of architecture
  - A formal approach to organizing IT resources is still a relatively new practice

- Just how big is the big picture?
  - Architects must have an enterprise-wide view

- Where are the architects?
  - It’s hard to learn architecture at college – most learn on the job

Where Do You Find Enterprise Architects?

- Enterprise Architects (EAs) must have the big picture of the relationship between business & technology

- Some are more technical, some are more business-oriented, many organizations put both types together on a team

- Typically rise through ranks internally, because of need for intimate knowledge of business

- Supplement EA team with consultants, but don’t let consultants drive EA
Building the right SOA team

• Shared Services cross organizational boundaries

• Siloed IT management styles are becoming obsolete

• The new role for enterprise architects

Building Support for SOA

• Find your champion
  – May be LOB manager, CIO, management-level architect, or other architect

• Build the business case
  – Solve business problems while transitioning to new architecture

• Tackle project iteratively within context of overall plan
Challenge: Inertia in the Organization

- Architecture doesn’t have features and business executives pay for features!
- Moving to SOA means breaking down silos and sharing resources
- The technology change is easy – it’s the human change that’s the hard part!

Change Management Issues

- Organizational change more challenging than technological change
- Keep business focused on TCO
- Focus on human aspects of change management (education, etc.)
Interaction Challenges

Services blur the Application / Network Boundary!

- Cultural Issues
  - Network Ops and Developers don’t talk to each other

- Budget issues
  - Who pays for Service Infrastructure / intermediaries?

- Responsibility issues
  - Who is in control of policy?

More Interaction Challenges

Architecture is difficult to mandate

- Management issues
  - People tend to avoid risk, stay within “comfort zone” - may appear stubborn

- Technical issues
  - Architecture is a difficult subject

- Cultural issues
  - The “Ivory Tower” problem...
The “Ivory Tower” Problem

- Architects create design and other artifacts, but don’t have the authority or mandate to require their use
- Development team considers them optional
- Business likes idea of architecture in principle, but short-term needs trump best practices
- When architects are external consultants, the “not invented here” syndrome makes the Ivory Tower worse

Convincing Technical Specialists

- Among the most risk-averse are technical specialists – mid to late-career experts in a (typically legacy) technology (e.g., “COBOL Jockeys”)
- Architectural change threatens their careers
- Solution:
  - Work with younger developers to build acceptance for SOA (eventually the TS’s will come around)
  - Take a “leave and abstract” approach over “rip and replace”
Working with IT Middle Management

- Middle managers threatened by SOA because of the Service domain reorganization

- Solution:
  - Technical specialties still required
  - New opportunities for Service domain management

The SOA Team

- Business analysts/business process architects
  - Define Service specs depending on business requirements
  - Decompose and recompose processes

- Enterprise/SO architects
  - Codify policy and best practices
  - Create Service model

- Technical specialists/project architects
  - Specify implementation

- Service provider/consumer developers
  - Implement requirements and policy

- Testers / Quality Assurance
  - Insure conformance, simulate Service providers & consumers

- Network, operations and security personnel
  - Contribute relevant expertise to project
Bring Together Different Mindsets

- **Developer Mindset:** “Bottom-Up”
  - Everything is a Service or an Interface
  - Goal: connect Services
  - Method: Use objects and App Servers
  - Problem: Too many things to connect!

- **Business Mindset:** “Top-Down”
  - Everything is a Process
  - Goal: Run business efficiently: manage processes
  - Method: Use diagrams and flowcharts
  - Problem: How can you turn “shelf-ware” into software?
Project Management for an SOA Project

- Pilot project much like a standard IT project, because business Services not yet in place.
- As your SOA matures, you must shift to a more agile, model-driven approach that requires more flexible project management.
- Basics of project management won't change (resource management, client management, schedule/dependency management).
- Project managers will have to deal with larger, more diverse teams.
- Maintaining agility requires the project manager to change the project plans over time more frequently.
- Key to keeping SOA projects on track: measurement of key indicators
  - Quality indicators
  - Governance indicators
  - Other indicators

The Relationship with Portfolio Management

- **SOA rollout** projects vs. ongoing **Service lifecycle** projects
  - Break up SOA rollout into individual projects, based on iterative approach showing business value at each step
- Plan for ongoing change
  - New Services
  - New versions of Services
  - Ongoing reconfiguration of SOBAs, policies, etc.
Challenge: Service Granularity

- The trick of building composable Services is building at the right level of granularity

- Challenges:
  - Engraining business logic into code
  - Decomposing legacy services that are not fine-grained enough

- Method
  - Top-down process decomposition, vs. bottom-up Service development
  - Must be iterative

Process Decomposition

- Identify processes within scope
- Break them down into subprocesses with eye for redundancy
- Drive to proposed list of business Services based upon potential reuse
Challenge: Reuse = Sharing

We all learned to share in kindergarten...

But by the time we get to the working world, we forget how!

Planning for Reuse

- Process decomposition → areas of redundancy
- Business requirements for visibility & efficiency → areas of data/functionality overlap
- Legacy modernization → refactoring of current functionality

Governance of reuse should be in governance framework
Code Reuse vs. Service Reuse

• Reuse the old way: code reuse
  – Reusable code libraries and subroutines date back to the earliest days of computing – the “Holy Grail” of programming
  – Every developer wants to make changes, branching the code base, reducing reusability
  – Hard to write reusable code, as requirements are never clear

• Reuse the new way: Service reuse
  – Reuse at runtime based upon contracted functionality
  – Loose coupling leads to flexible reuse
  – Appropriate governance essential!

The Power of the SOA Center of Excellence

• SOA experts who maintain a knowledge base of best practices
  – General and company-specific
  – Design time and runtime

• Drives SOA policy (either explicitly or implicitly)

• Can unify approaches across a large organization
Enabling Service Domains

- A Service Domain is a logical grouping of shared Services with a common business context.
- Examples: customer-facing Services, purchasing-related Services.
- Manage Services by managing the Domains.
- Move away from traditional IT silos for the purposes of managing Services, but retain technical teams as needed.

Challenge: People, Change and Fear

- People are inherently resistant to change.
- People consider job security, authority and responsibility when asked to share.
- Fear is the strongest emotion of all!
Thank You!

ZapThink is an advisory, analysis, & influence firm focused exclusively on Service-Oriented Architecture, Web Services, & Enterprise Web 2.0.


Jason Bloomberg  
jbloomberg@zapthink.com

Ronald Schmelzer  
rschmelzer@zapthink.com