



# SOA Pilots

Jason Bloomberg & Ron Schmelzer  
ZapThink, LLC

Take Credit Code: NOPILOTS

Copyright © 2006, ZapThink, LLC

zapthink <sub>1</sub>



zapthink

## Phases of SOA Maturity

- Taking an iterative approach to SOA is a fundamental best practice
- How do you avoid fragmenting your SOA efforts?
- How do you keep the “big picture” in mind through as the SOA implementation matures?

Copyright © 2006, ZapThink, LLC

2



## Getting Started with SOA

- Identify the business problems
- Determine the overall scope of SOA initiative
- SOA is no silver bullet – target problems that SOA is particularly good at solving
  - Problems with heterogeneity
  - Problems with inflexibility
  - Problems with lack of visibility



Copyright © 2006, ZapThink, LLC

3



## 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



Copyright © 2006, ZapThink, LLC

4



zapthink

## SOA Pilots

- A few high ROI Services
- Build acceptance for SOA
- Get team up to speed
- Work out the kinks
- Pilot the governance model
- Part of an iterative approach to SOA



### **DANGER! Avoid the SOA Pilot Pitfall**

- Piloting only the *Services* instead of the *architecture*
- Remember, the pilot is one step on the roadmap

Copyright © 2006, ZapThink, LLC

5



zapthink

## Mission-Critical SOA

- Services incorporated into core business processes
- Issues of security, management, granularity resolved
- Provides ongoing agility – built to change

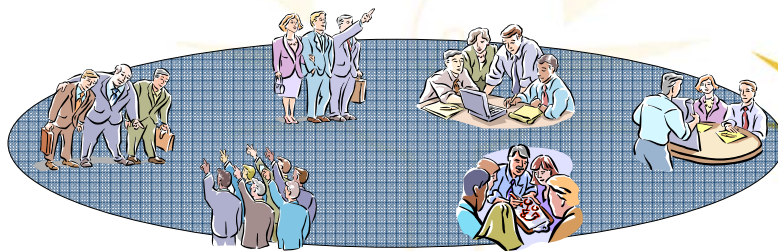


6



## Cross-Departmental SOA

- Organizational issues of governance and control become paramount
- Long-term architectural plan critical
- Increased focus on semantic issues



Copyright © 2006, ZapThink, LLC

7



## Enterprise SOA Buildout

- The corporate enterprise architecture becomes Service-Oriented
- Service lifecycle becomes dominant framework for IT change
- Service abstraction frees company to retire legacy as appropriate





# The Service-Oriented Enterprise

- IT resources are available on demand to businesses as Services
- The Service-oriented abstraction layer enables companies to run their operations and conduct business with each other in a dynamic and automated fashion
- Business drives IT, and agile IT enables agile businesses



Copyright © 2006, ZapThink, LLC

9



# SOA Maturity Model: HP




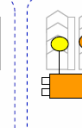
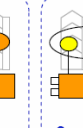
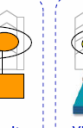
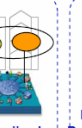
		SOA Maturity Levels				
		Level 1 Ad-hoc	Level 2 Basic	Level 3 Standardized	Level 4 Managed	Level 5 Adaptive
SOA Domains	Business	Minimal business interest in SOA	Business is aware of SOA	Business generally complies with SOA	Business proactively supports SOA	SOA is fundamental to business operations
	Program mgmt.	SOA is project focused	SOA efforts are business unit based	SOA is federated, but not integrated	SOA is integrated at corporate level	SOA is enterprise-wide and extends to partners
	Governance	Some acknowledgement of governance issues	Some governance processes, individual responsibility	Governance guidelines defined and integrated into process	The value of governance is fully understood	Advanced understanding of IT governance
	Architecture	Limited or ineffective architecture	Architecture program exists, and architecture is defined	All IT initiatives comply with the architecture	Architecture is business driven and is auditably linked	Architecture and business are executed as integrated
	Operations & mgmt.	No management of services, infrastructure elements only	Management of applications and infrastructure in terms of SLAs	Management of business services	Proactive management of business services linked to component services	Management of business services integrated into business operations
	Supply and demand	Business needs are all met using technology components	All services are provided internally	Value based sourcing decisions	Services sourced from multiple providers	Dynamic service sourcing from multiple sources
	People	Staff have little or no knowledge of SOA	Understanding of SOA is limited to IT management & architects	SOA education is required for all IT staff	Ongoing SOA education is attended by all staff	SOA is embraced by all staff and actively promoted
	Enabling technologies	There is no service infrastructure in place	SOA infrastructure is limited to exposing functionality as services	Standardised enterprise-wide SOA infrastructure	Large-scale managed SOA infrastructure	Integrated, dynamic SOA infrastructure

Source: HP



zapthink

# Service Integration Maturity Model: IBM

	 Silo	 Integrated	 Componentized	 Services	 Composite Services	 Virtualized Services	 Dynamically Re-Configurable Services
Business View	Function Oriented	Function Oriented	Function Oriented	Service Oriented	Service Oriented	Service Oriented	Service Oriented
Methods	Structured Analysis & Design	Object Oriented Modeling	Component Based Development	Service Oriented Modeling	Service Oriented Modeling	Service Oriented Modeling	Grammar Oriented Modeling
Applications	Modules	Objects	Components	Services	Process Integration via Services	Process Integration via Services	Dynamic Application Assembly
Architecture	Monolithic Architecture	Layered Architecture	Component Architecture	Emerging SOA	SOA	Grid Enabled SOA	Dynamically Re-Configurable Architecture
Infrastructure	Platform Specific	Platform Specific	Platform Specific	Platform Specific	Platform Specific	Platform Neutral	Dynamic Sense & Respond
	<b>Level 1</b>	<b>Level 2</b>	<b>Level 3</b>	<b>Level 4</b>	<b>Level 5</b>	<b>Level 6</b>	<b>Level 7</b>

Copyright © 2006, ZapThink, LLC

Source: IBM 11



zapthink

# Key Elements of Successful SOA Projects

- Building the right SOA team
  - A diverse group that bring different perspectives and wide-ranging support to the implementation
- Handling organizational/people issues
  - Human resistance to change more challenging than the technical issues
- Tackling the project iteratively
  - Won't have full spec as you get started
- Managing the Service lifecycle
  - Very different from the traditional software development lifecycle (SDLC)

Copyright © 2006, ZapThink, LLC

12



## Building the right SOA team



- Shared Services cross organizational boundaries
- Siloed IT management styles are becoming *obsolete*
- The new role for enterprise architects

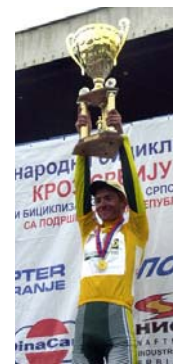
Copyright © 2006, ZapThink, LLC

13



## 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



Copyright © 2006, ZapThink, LLC

14



zapthink

## Change Management Issues

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



Copyright © 2006, ZapThink, LLC

15



zapthink

## Interaction Challenges

**Services blur the Application / Network Boundary!**



Developer/Architect



Network Operations

- **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?

Copyright © 2006, ZapThink, LLC

16





zapthink

## More Interaction Challenges

### *Architecture is difficult to mandate*



Architect



Development/Testing

- **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...

Copyright © 2006, ZapThink, LLC

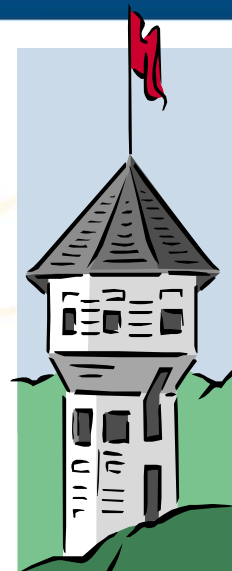
17



zapthink

## 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



Copyright © 2006, ZapThink, LLC



## 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”



Copyright © 2006, ZapThink, LLC

19



## 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



Copyright © 2006, ZapThink, LLC

20



## Implementing SOA: Bottom-Up

*Put Service wrappers around existing applications*



- Pros:
  - Reduces cost of integration
- Cons:
  - May not be reusable
  - May be redundant
  - Management challenges

Copyright © 2006, ZapThink, LLC

21



## Implementing SOA: Top-Down

*Create architectural plan & detailed design*

- Pros:
  - Agility, reuse, flexibility
- Cons:
  - May not be implementable
  - Difficult to budget



Copyright © 2006, ZapThink, LLC

22



## SOA *Must* be Both

- Develop the vision (but not the details) ahead of time
- Decompose some processes to identify target Services
- Build modest set of Services
- Compose applications to enable flexible processes
- Refine architectural plan
- Repeat

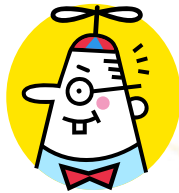
*SOA should be iterative*

Copyright © 2006, ZapThink, LLC

23



## 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?

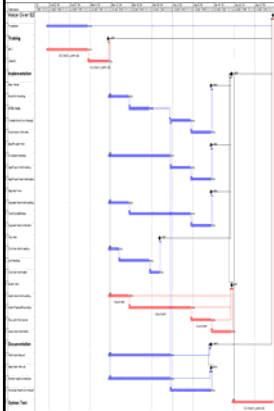
Copyright © 2006, ZapThink, LLC

24



zapthink

## 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

Copyright © 2006, ZapThink, LLC

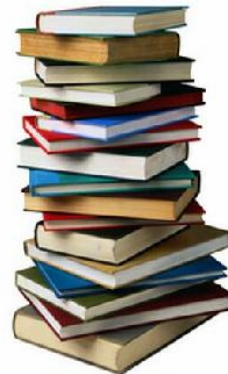
25



zapthink

## 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.



Copyright © 2006, ZapThink, LLC

26



# 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

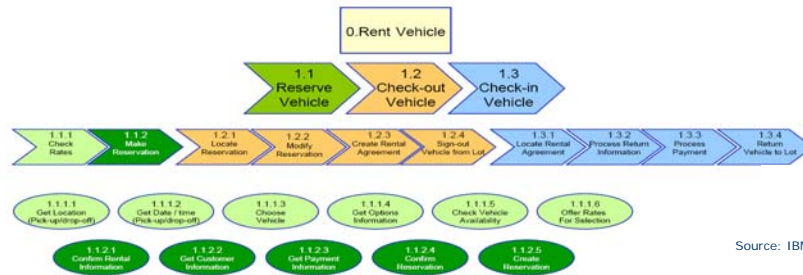


Copyright © 2006, ZapThink, LLC

27



# Process Decomposition



Source: IBM

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

Copyright © 2006, ZapThink, LLC

28



## 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**

Copyright © 2006, ZapThink, LLC

29



## 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!



Copyright © 2006, ZapThink, LLC

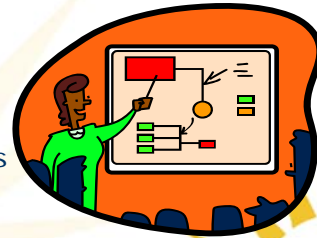
30



zapthink

## Architectural Visioning Session

- Attendees: architecture team, business analysts responsible for business processes
- Provide the SOA perspective
- Processes drive the Services, Services drive the technology
- Split into two initiatives: process definition followed by Service definition
- Balance "big picture" with realistic pilot starting point



Copyright © 2006, ZapThink, LLC

31



zapthink

## The Governance Role of Architecture

- The architecture team should drive IT governance
- SOA initiatives should coordinate with governance initiatives
- Approach to governance should be Service-oriented

Copyright © 2006, ZapThink, LLC

32





zapthink

## 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

Copyright © 2006, ZapThink, LLC

33



zapthink

## 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



Copyright © 2006, ZapThink, LLC



## Selecting the right team: Consulting Partners



- 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 have in-house EA talent

Copyright © 2006, ZapThink, LLC

35

# Thank You!



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

Read our new book, *Service Orient or Be Doomed! How Service Orientation Will Change Your Business.*



Ronald Schmelzer  
rschmelzer@zapthink.com



Jason Bloomberg  
jbloomberg@zapthink.com

Copyright © 2006, ZapThink, LLC

zapthink

36