



# Zen or Tao of SOA & Software As A Service

---

David Moskowitz  
Productivity Solutions, Inc.





# Agenda

---

- Common vocabulary
- When to use / not use SOA
- When to use / not use SaaS
- What's real, what can you expect?
- Q & A



# Agenda

---

- **Common vocabulary**
- When to use / not use SOA
- When to use / not use SaaS
- What's real, what can you expect?
- Q & A



# Ground Rules

---

- *Service* is "polymorphically overloaded" 😊
  - SOA != SaaS ; SaaS != SOA
- The concept of Service in SOA is not the same as the concept of Service in SaaS
  - SOA is about business process
  - Agility & business requirements
  - Business driven to improve responsiveness and align IT with business goals
- SaaS is about reducing costs – usually but not always involving a 3rd party vendor



## 3 Views of *Service*

---

- SOA maps service to business process
- SaaS maps service to revenue
- ITIL maps service to **customer** value and **customer** outcomes
  - A means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks.
  - *Service Management* is a set of specialized organizational capabilities that provides value to customers in the form of services.



➤ See "What Is ITIL" on the CSS2007 CD



# SOA Common Vocabulary

---

- Service: reusable bit of s/w that can be invoked *via* a published interface
  - Service contract
  - Has a recognizable business function that plays a clear role in multiple applications
- Service reuse: a major payoff point of SOA
  - Purpose: decrease time to market, reaction time, and increase business flexibility and adaptability
  - How do you define re-use and reusability?
- Service broker(ing)
  - Part of reuse
  - Publish/advertise service contracts



# SOA is an Architecture

---

- Architecture defines system components & how they interact (100K ft level)
  - Components aren't any specific type of object
  - Abstract "modules" deployed as one or more units on one or more servers
- Architecture also defines the externally visible properties and their relationships.
  - Structure, relationships and patterns
  - Defines expectations expressed as a contract



# Service Oriented Architecture

---

- SOA is neither a technology nor an implementation
  - SOA formally separates interface & implementation
    - Service-consumer views service as an endpoint
    - Supports contract with defined request & response messages
    - Dictates a Find Bind Execute (FBE) paradigm
  - Web Services take this one step further -- define a formal mechanism for the FBE process
- WS is something that can be described by WSDL





# SOA loosely coupled services

---

- SOA expresses the definition of loosely coupled software "pieces" that model or map business services, processes, or problem space.
  - The driver is the business process
  - It's the business process that is the major driver not the technology
    - Doesn't mean the technology isn't important (that's another topic for either the fireplace or the Q & A).
- Ultimate goal is more agile and adaptive IT



# SOA beginnings

---

- SOA: not a technology
- SOA: built on standards-based protocol stack
  - ... but not limited to the stack
- SOA is also about business re-engineering
  - SOA isn't about applications, it is about identifying business processes that can be expressed as one or more (connected) services.
  - BPOA (Business Process Oriented Architecture)



# SaaS

---

- Software as a Service (SaaS) targeted primarily at mid-market as a way to get larger enterprise-like services.
- Requires knowledgeable people to implement, migrate, and manage.
  - Doesn't put the IT guy out of business, shifts the skills set, doesn't eliminate the need.



# SaaS

---

- Instead of funding a huge software project, the small business owner rents time on a standardized system.
  - Costs shared by multiple (thousands?) users.
  - No up-front licensing fees.
  - Instead of buying software, you run web-based software.
  - PC becomes the terminal, and your service provider acts as the "mainframe."



# SaaS Acronyms

---

- ASP: Applications Service Provider
  - Usually non-critical IT functions, organization doesn't want the overhead
  - Existing internal software compatibility minor
  - Oracle On-Demand ERP
- BPO: Business Process Outsourcer
  - Handles entire process
  - Best when **how** the vendor does "it" is irrelevant
  - ADP payroll processing



# SaaS Acronyms

---

- **MSP: Management Service Provider**
  - Set of IT services (e-mail security)
  - Business services (handling telecom expenses)
  - Both business and IT services (call center)
  - Works when the MSP provides services outside the core competency of the organization.
  
- **MTA: Multi-tenant Architecture**
  - A different form of hosting that provides a single instance of the software.
  - Cost of maintenance shared by multiple users
  - This is the primary context for discussion in this session



# INATT

---

- It's Not About The Technology...
  - It's not about the technology, alone
- Either statement misses the point.
  - Opposite does, too: it IS about the technology
- Technology is the enabler, but should not be the starting point for the discussion.
  - Don't do SOA or SaaS because of technology, do it because it's the right business decision
  - Technology should not be the driving factor



# SOA or SaaS as Part of Web x.0

---

- First identify what a Web version is?
  - Define the line of demarcation between versions
  - Can talk about changing patterns of use & evolving technology (*kaizen*)
  - Is that really a new version?
- SaaS is usually Web-based so there is an argument in it's favor
- SOA is about BPOA that may or may not involve Web-based technologies
  - For example, SOA wrappers around existing (legacy) systems





# SOA / SaaS & the Web

---

- The Web becomes a way of dealing with a UI to a set of "services" (WWR – Warning Will Robinson... )
- SaaS may become part of an SOA implementation as part of governance, or outsourced services, or...
- SOA is not likely to be part of SaaS
  - Partial start of the confusion



# Governance

---

- Governance is a critical part of SOA
  - Everyone knows the term until...
  - "...difference between knowing the name of something and knowing something."  
(Richard Feynmann)
  - Not just who, what, when, & where; need both why & how, too.
- Wikipedia: "Governance makes decisions that define expectations, grant power, or verify performance."
- IT Governance: Management system that provides authority & control over every aspect of the IT service strategy & lifecycle



# Agenda

---

- Common vocabulary
- **When to use / not use SOA**
- When to use / not use SaaS
- What's real, what can you expect?
- Q & A



## SOA when you're willing to...

---

- Take time to understand & evaluate the underlying business issues.
- Start with the business process(es) so that the solution to the problem is defined, cast, and considered in terms of the way business process(es) work (or should work) together.
- Provide appropriate governance



# Governance and SOA

---

- Governance is not just about code, it's also about communication and process.
- Good governance is also about distributed responsibility to teams that have budget, authority, and capability to maintain and modify each service!
- Governance is part of the process
  - If the organization can't (won't) pay attention to this aspect, skip SOA



# You're not ready for SOA if...

---

- The initial thinking is about the technology
- Difficulty defining the business processes (BP) or expressing the "solution" in BP terms
- Organization unwilling to consider changes to organization structure or business process
- Not ready or capable of providing the proper communication required by good governance
- There isn't a responsible knowledgeable champion to keep the SOA process on track
- There are 8 anti-patterns

➤ <http://www.softwaresummit.com/2006/speakers/MoskowitzPatternsAndAntipatterns.pdf>

# The Real SOA Problem

No change



== No gain!



Source: WSJ 7/2/2007



# Why SOA, the Take-away

---

- Starts with the business process first.
- Not driven by technology
- **Is** driven by one or more changes at the business level before considering a technology solution.
  - Yes, it makes sense to ask, "Can we do this using technology?"
  - Business == Driver
  - Technology == Enabler
  - Some SaaS may be part of the solution space





# SOA Isn't Easy or Trivial

---

- SOA is neither a quick fix...
  - Nor a short term project.
- It takes time to understand the business issue involved.
  - Potential anti-pattern: If everyone agrees in "2 minutes" that this is the problem, "Yep we've got the business process defined, design and implementation starts tomorrow."
  - Another anti-pattern... we've got this nifty new tool, so we'll do SOA today.



# Agenda

---

- Common vocabulary
- When to use / not use SOA
- **When to use / not use SaaS**
- What's real, what can you expect?
- Q & A



# SaaS is...

---

- Single code base is used by multiple users over a network (including the Internet).
  - Users may configure aspects of the experience
  - Change to the software is available to every user.
- Underlying data model and architecture fixed
  - Advantage 1: minimal time managing/fighting compatibility issues.
  - Advantage 2: Support costs reduced, software runs on known hardware
  - Advantage 3: Very rapid deployment



# SaaS is...

---

- Also a business for third party vendors
  - Typically costs less than either in-house development or up-front licensing fees (add the costs of support and the advantages tips even more toward SaaS).
    - *e.g.*, Salesforce.com, ADP Payroll, Echopass
- May be bundled with BPM for best practices
- MTA == Multi-tenant architecture
  - One instance, many concurrent users
  - Little maintenance on the user side



# On Demand: MTA or STA

---

- On demand is Web-based solution to a specific problem.
- MTA single instance, multiple users
- STA one instance per user (or organization)
- The more money you have, the more you care about customization and capability
  - ...making SaaS potentially less appealing, from the perspective of customization not money.
  - Money creates potential (or more) options



# MTA Advantages

---

- For purposes of this presentation the term vendor is applied to both someone selling SaaS type service as well as an in-house deployment that is on-premises.
- Vendor:
  - one instance overhead
  - In-house has fewer security issues than external
  - In-house makes sense if there are multiple UIs (or views) required for the same system (shop floor, scanning, PoS systems, *etc.*)



# MTA Advantages 2

---

- User:
  - little maintenance
  - Accommodates price sensitivity – particularly if price limits exceed functionality requirements.
    - Provided what you need is within the scope of the MTA application
- Variant: STA – one instance and your own database hosted on their servers.... Costs more than MTA but less than individual licenses.



# When to consider SaaS

---

- If the "it" isn't complex or is vanilla
  - Unless there's a reason to build
  - For example: service levels vendor can't (won't) guarantee
- The software has to do what you need.
- Don't have (or don't want) a development organization to implement and maintain
- Need to explore best practice solution without the internal expertise





# Avoid External SaaS if...

---

- Almost everything must be customized
- "It" touches the core of the enterprise
  - Applications that touch (or provide) the differentiators between competitors
  - *e.g.*, BI, ERP, financial, manufacturing systems...
- The functionality is so critical to operations that the organization must own them
- High number of integration or touch points to other software



# Agenda

---

- Common vocabulary
- When to use / not use SOA
- When to use / not use SaaS
- **What's real, what can you expect?**
- Q & A



# Case Study: SOA Starts as BPOA

---

- Company starting concept: improve customer service, loyalty, and satisfaction.
- Made business process changes, first.
  - Change CIO responsibility to include CCO
- Completed 4-year BPOA project
  - Created single lifecycle view of each major item sold.
  - Tracked from the instant order placed through manufacturing, testing, transportation, delivery, use, routine service, repair, etc...
  - Also included responsible customer-side people
- Net result (1st year): customer sat +60%, loyalty +51%, customer service costs -71%, profitability (net SOA development costs) +37%



# Case Study: SaaS

---

- MSP: outsourced travel services: businesses with \$250k+ annual travel budget
  - MSP has contracts with almost every major airline and hotel chain world-wide.
- Customers contract for a specific number of room-nights and flights
  - Customers get contract compliance monitoring, travel security advisors, and best practices.
  - Also customized reporting, contract negotiations
- Service used by large & small organizations
- Savings vary by organization size & budget



# Duh comments of the day

---

- Neither SOA nor SaaS is the panacea for every problem.
- A vendor has an axe to grind with respect to other approaches



# Real v Hype

---

- Neither is a universal panacea
  - Both SOA and SaaS are real and work in the correct situations and environments.
- Some discussion that SOA is INATT
  - Both sides miss the point. SOA is about business 1st; technology is both the enabler and the implementation vehicle.
- SaaS has potential for multiple organizations IF...



# Some additional thoughts

---

- Solutions should be driven by business not just technology
- The right business solution may include both SOA and SaaS
- They aren't competing technologies
  - They're meant to solve different problems



# Final Word on Service

---

- SaaS service is about delivery of automated processes & functionality
  - Think about mash-ups
- SOA service, to make it explicit, is also about organization culture as it applies to **service lifecycle**.
  - Deliver value to customers by facilitating outcomes customers want without ownership of the costs and risks.



This is also the ITIL definition of *service*





# Questions ? ? ?

---

**If you  
don't ask,  
who will?**

**If not now,  
when?**



**There  
aren't any  
dumb  
questions.**

**The only dumb  
question is the  
one not asked!**





# Thank You

---

For more information:

David Moskowitz

Productivity Solutions, Inc.

147 Ashland Avenue

Bala Cynwyd, PA 19004

+1-610-726-9925

[davidm2@usa.net](mailto:davidm2@usa.net)

SkypeID: davidmosk

