

## What's the Big Deal about Service-Oriented Architectures?

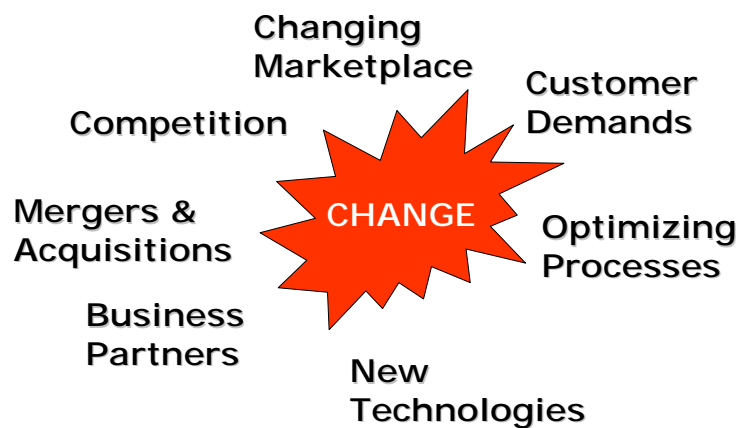
**Jason Bloomberg**

**Senior Analyst**

**ZapThink, LLC**



## Business Constant: Change



***A Business is Never STATIC***





## IT: Fulfilling Business Requirements

### Business Requirements

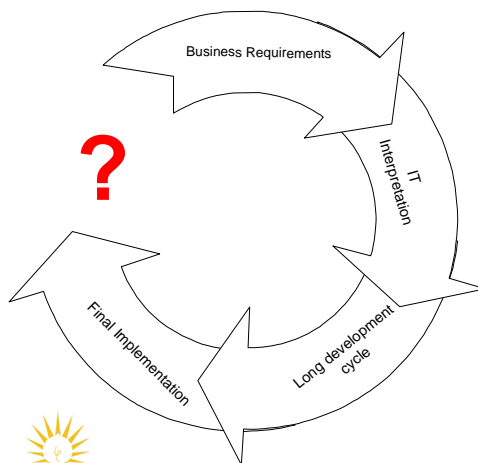
- Service Customers
- Manage Operations
- Increase Worker Productivity
- Communicate with market
- Ensure reliable and secure operations
- Develop new products and services
- Respond to new business drivers

### IT Capabilities

- Implement CRM Systems
- Implement ERP Systems
- Manage desktop environments
- Manage server environments
- Manage email systems and web sites
- Manage network and storage operations
- Develop applications



## However, it rarely works that way...



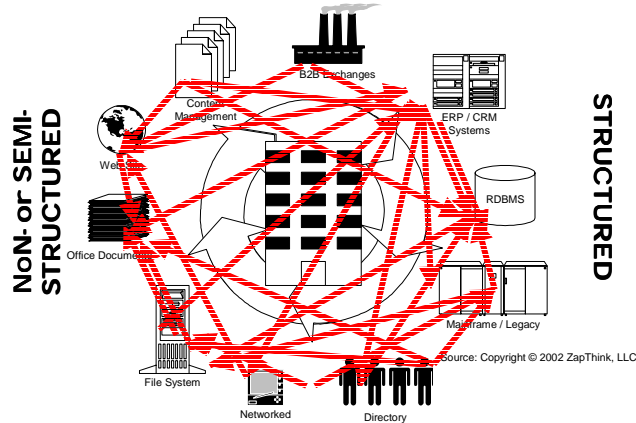
- Requirements change
- Interpretations often inaccurate or limited
- Lengthy development cycles impervious to change
- Implementations “cast in concrete”

**Result: IT that places limitations on Business**



## The Integration Challenge...

### The N-squared Integration Problem:

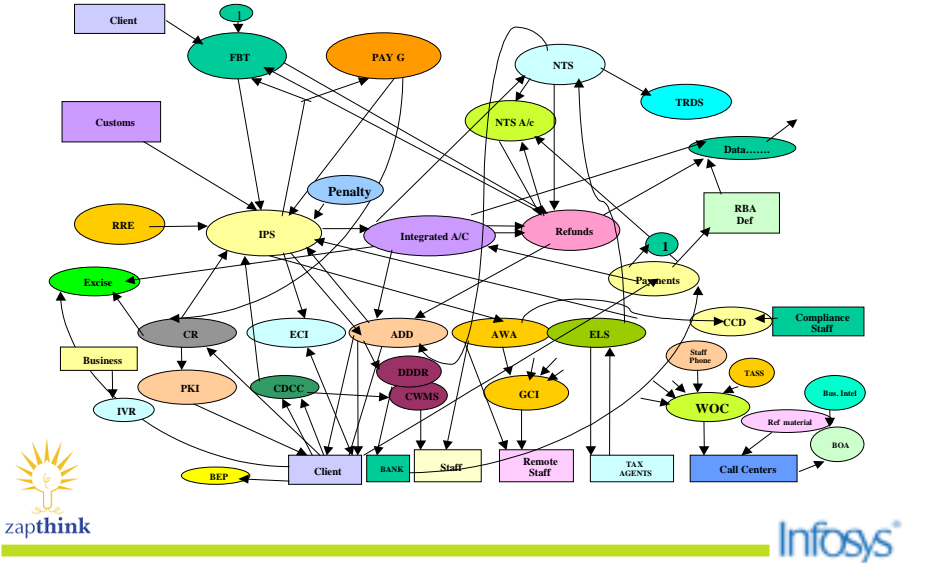


## Integration Approaches of Yesterday

- Custom Integration: Coding to Interfaces
  - APIs: COM, Java, COBOL, Assembly?
  - Distributed Computing?: DCOM, CORBA
  - Screen-Scraping and Emulation (3270 and HTML)
  - Message-Queues
- EAI and B2Bi Middleware
  - Automating interface-level integration
  - Bus or hub-and-spoke architecture

Fundamentally *brittle* approaches to integration

## The "Rat's Nest" Architecture



## What is a Service-Oriented Architecture?

- Access software via Services that are easy to find and connect to
- Web Services provide a *standard* way of building and accessing Services
- Users can build applications out of Services



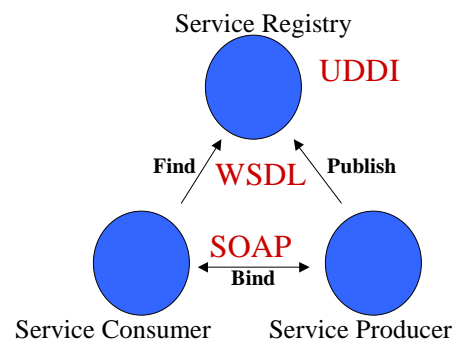
## Have We Been Here Before?

- 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

*Standards-based*  
interfaces to  
software  
functionality



## Web Services in the Present...

### Web Services are in the horseless carriage phase

- Where new technology is applied in the patterns of the earlier technology
- Web Services are used to simplify integration



## Web Services in the Future...

### New approaches to software development, engineering, architecture, and management



## Web Services are the Trees....



## Web Services *Idées Fortes*: Loose Coupling

- Consumer and Producer controlled by different people
- Changing one doesn't break the other
- Build one without being aware of the other



## Web Services *Idées Fortes*: Asynchrony

- The Web is *synchronous*: click a button and wait for a response
- Web Services can also be *asynchronous*: allow for long-running processes
- SOAs should be event-driven



## Web Services *Idées Fortes*: Coarse Granularity

- Business-oriented requests and responses
- Blocks of information exchanged
- Encapsulate APIs into fine-grained, atomic Services and compose them into coarse-grained, business Services

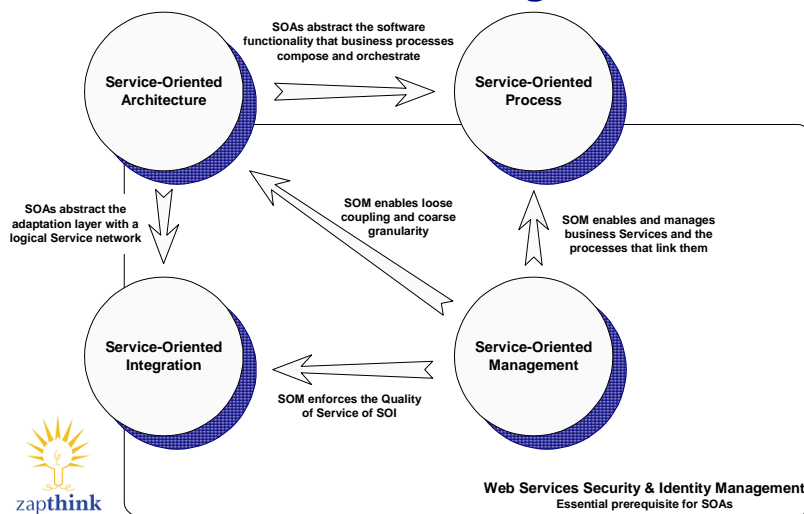




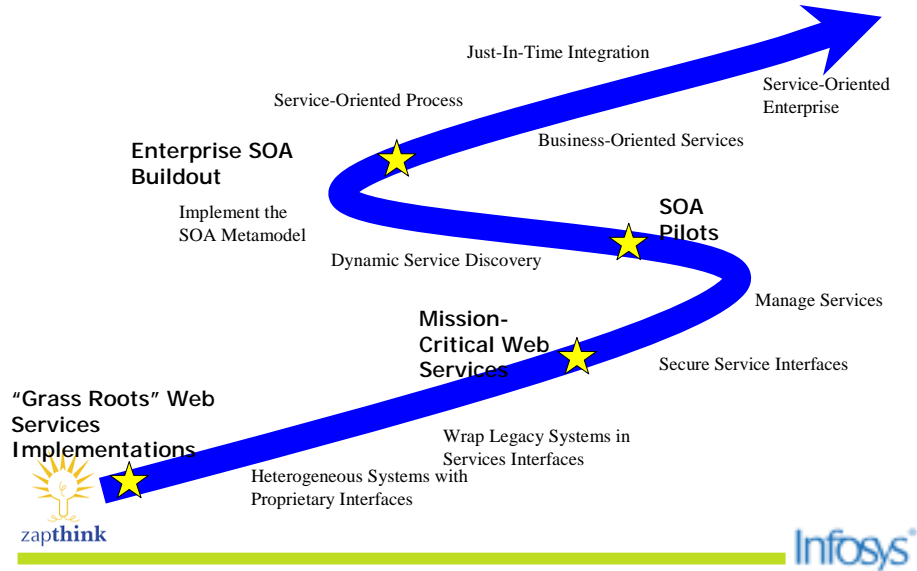
## Service orientation... the next big thing?

Approach	Timeframe	Programming Model	Business Motivations
Mainframe timesharing	1960s –1980s	Procedural (COBOL)	Automated business
Client/server	1980s-1990s	Database (SQL) and fat client (PowerBuilder, Visual Basic)	Computing power on the desktop
n-Tier/Web	1990s-2000s	Object-oriented (Java, COM)	Internet/eBusiness
Service orientation	2000s	Service-oriented (SOAP, WSDL, UDDI)	Business agility

## Achieving Business Agility Through SOAs

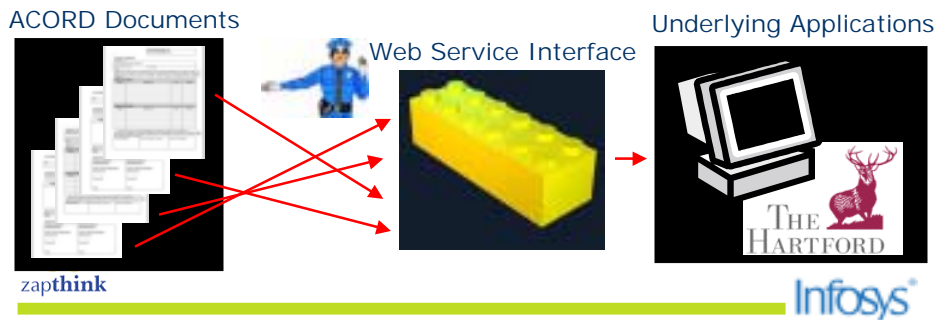


## The SOA Implementation Roadmap



## Case Study: The Hartford

- SO Business Application for insurance agents
- Services handle multiple versions of insurance forms
- SOA handles multiple versions of Services





## Case Study: Providence Health System



- SOA with 30 composite Services (each with ~10 atomic Services)
- Variety of uses, including patient portal, lab results, aggregation of medical records
- Supports HL7, X12 (payer interactions)
- Less duplication of effort, better patient care, faster & more complete patient information



## The Service-Oriented Enterprise

- IT resources are available on demand to businesses as Services
- The SOA provides an abstraction layer that enables companies to conduct business with each other in a dynamic and automated fashion
- Business drives IT, and agile IT enables agile businesses





ZapThink is an industry analysis firm focused exclusively on XML, Web Services, and Service-Oriented Architectures.

**Thank You!**

Take Credit for attending this presentation!

- Go to [www.zapthink.com/credit](http://www.zapthink.com/credit) and enter the code.
- Download a digital copy of the presentation
- Sign up for our ZapFlash newsletter
- Get a ZapCredit good toward free research and ZapGear!



Jason Bloomberg  
[jbloomberg@zapthink.com](mailto:jbloomberg@zapthink.com)

