OUCC 2015

Inspiring Innovation

Presentation:

Service Oriented Enterprise (SOE)

Presenter: Colin Bell (Director, Enterprise Architecture – University of Waterloo)

Date: May 4, 2015

Outline

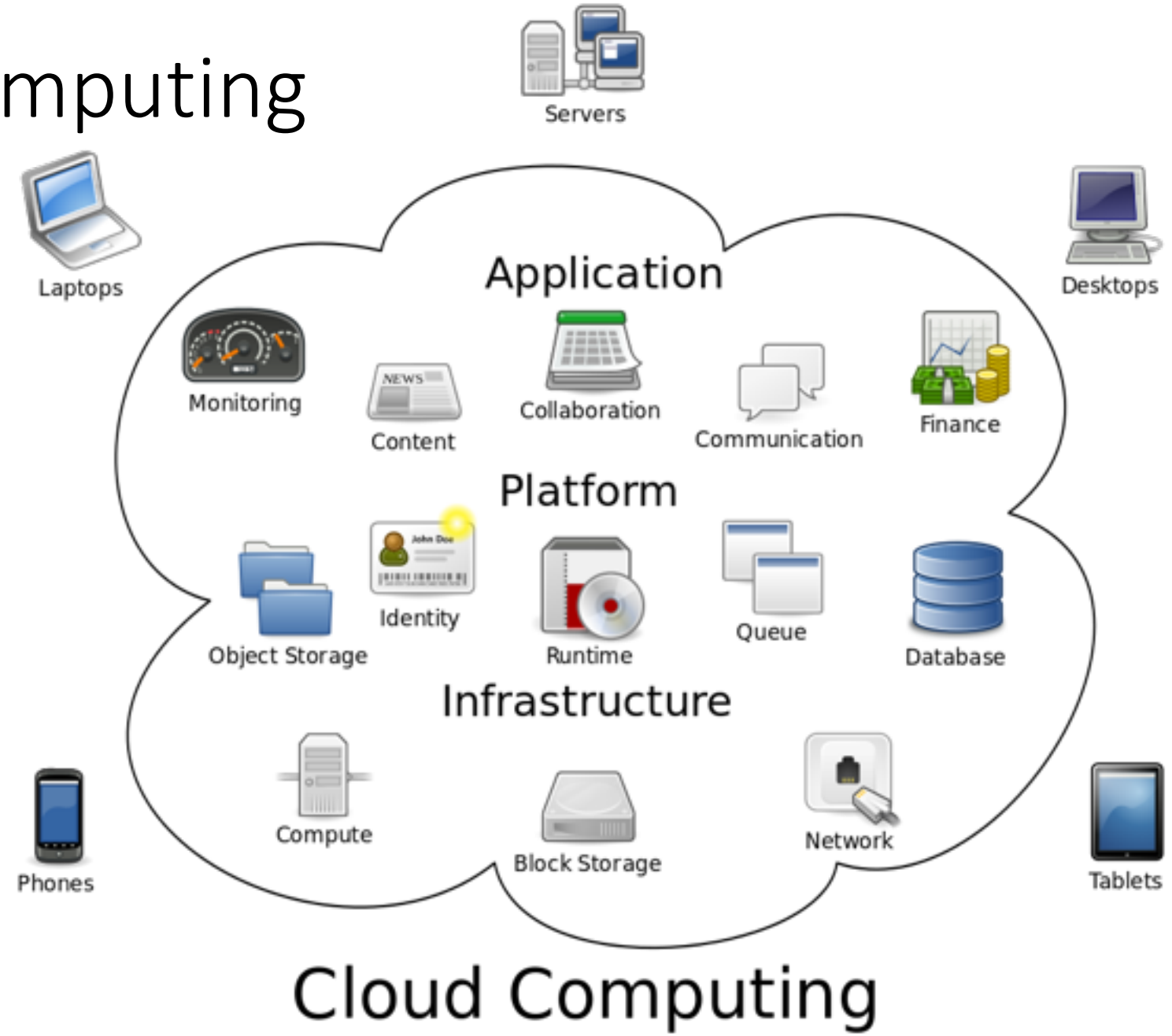
- Service Oriented Enterprise (SOE)
  - “The Cloud”
    - Service Delivery Models
    - Deployment Models
    - Economies of Scale
  - What is a Service?
    - Definitions
    - Practical Definition
    - Graphical Representation
- Building the Service Oriented Enterprise
  - Service Management w/ ITIL
  - Enterprise Architecture (EA)
    - Definition
    - Framework
    - EA BOK
    - Business Service Reference Models
  - Service-Oriented Architectures (SOA)

Definition: Cloud Computing

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation.

Source: http://en.wikipedia.org/wiki/Cloud_computing (2012)

Image: Cloud Computing



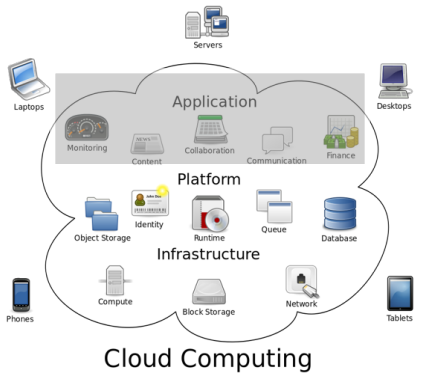
Source: http://en.wikipedia.org/wiki/Cloud_computing#/media/File:Cloud_computing.svg

Cloud Service Delivery Models

- Software (Application) as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

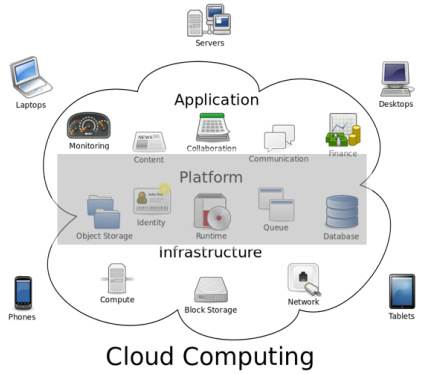
SaaS - Software (Application) as a Service

- **Traits:**
  - providers install and operate application software,
  - very little flexibility— you get what is provided, and;
  - users do not worry about underlying platform or infrastructure.
- **Examples:**
  - GMail / Google Apps
  - Hotmail / Microsoft Office 365
  - Salesforce
  - Desire 2 Learn / Brightspace



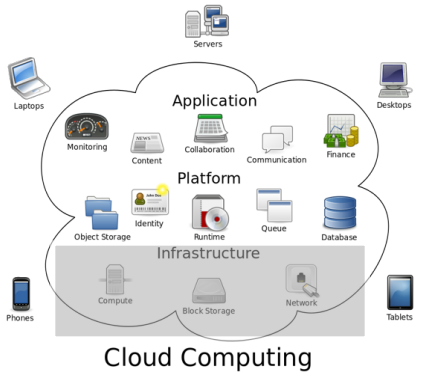
PaaS - Platform as a Service

- **Traits:**
  - provides users with an infrastructure pre-configured with a suite of tools,
  - often users are locked into a particular development suite, database, and Web server, and;
  - users can build and run software in a controlled environment.
- **Examples:**
  - Google App Engine
  - Engine Yard
  - Heroku



IaaS - Infrastructure as a Service

- **Traits:**
  - low-level access to basic computing components,
  - can choose own OS, software stack, and configuration settings, and;
  - clients are given their own virtual networks and data centre.
- **Examples:**
  - Amazon AWS
  - Microsoft Azure
  - Rackspace Cloud



Economies of Scale Benefits

SaaS > PaaS > IaaS

Why?

Less Flexibility + Fewer Features

⇒ Increased Specialization for Service Provider (decreasing per-unit costs)

⇒ Increased Prospective Customer Base for Service Provider (lower barrier to entry)

Cloud Deployment Models

- **Public Cloud**
  - Infrastructure that is owned by a corporation who sells their services to the general public.
- **Community Cloud**
  - Infrastructure that is shared amongst like-entities. Municipalities, Governments, non-Profit Organizations, and Non-Governmental Organizations often share these services.
- **Private Cloud**
  - Infrastructure that is operated solely for a single entity.
- **Hybrid Cloud**
  - A composition of two or more clouds that are separate at the lowest Infrastructure levels while allowing interconnection at higher levels.

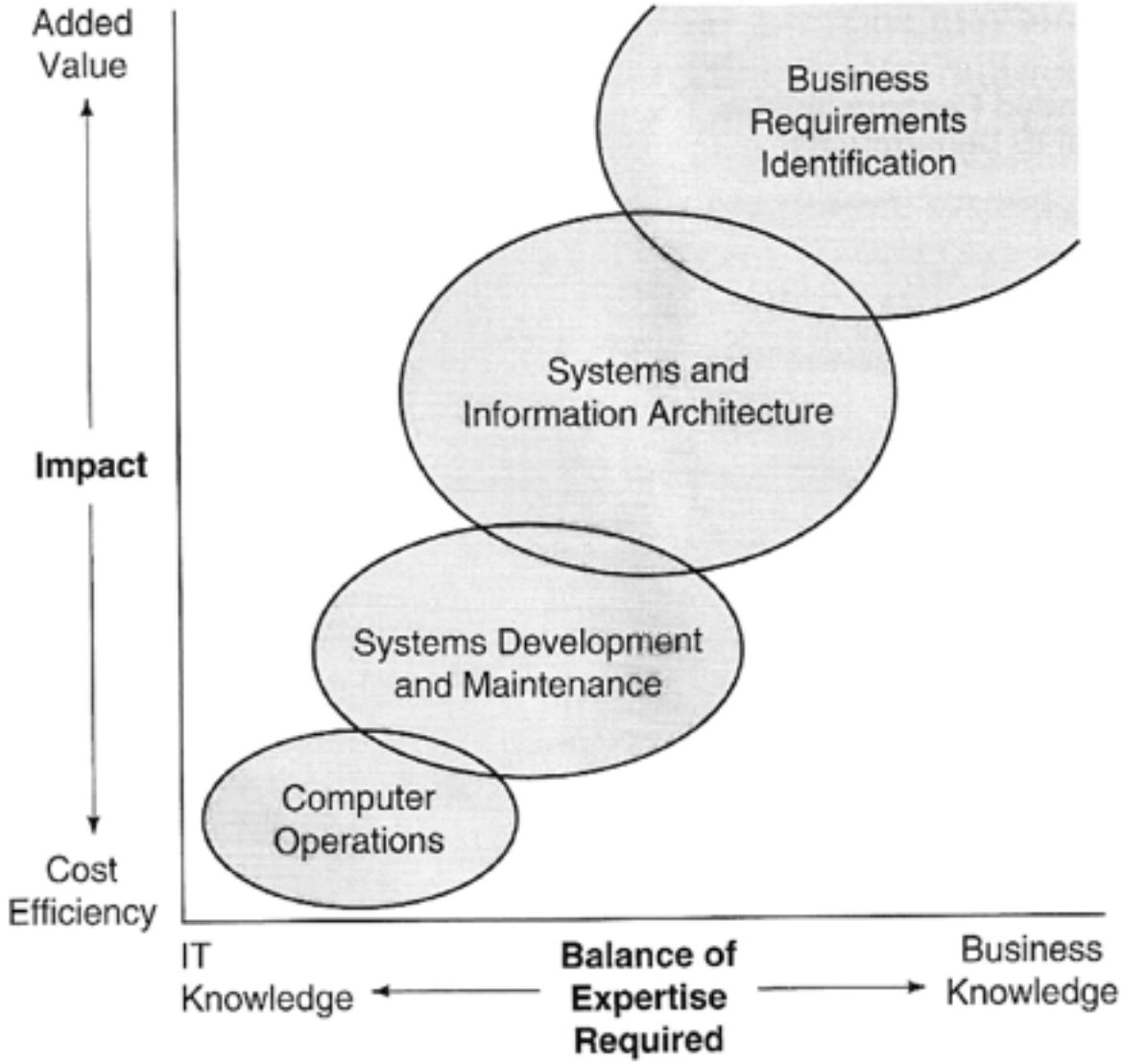
Value Generation (Impact++) vs. Cost

- By improving specialization, the cost of production (of services) can be driven down. By increasing the number of customers, revenue can increase as marginal costs decrease.
  - ***Economies of scale is kicking in.***

- When someone else can provide service for less, do we consider the Opportunity Cost?
  - ***Is maintaining the status quo a good idea?***

- What ’higher value’ things could we be doing to make the organization more productive?

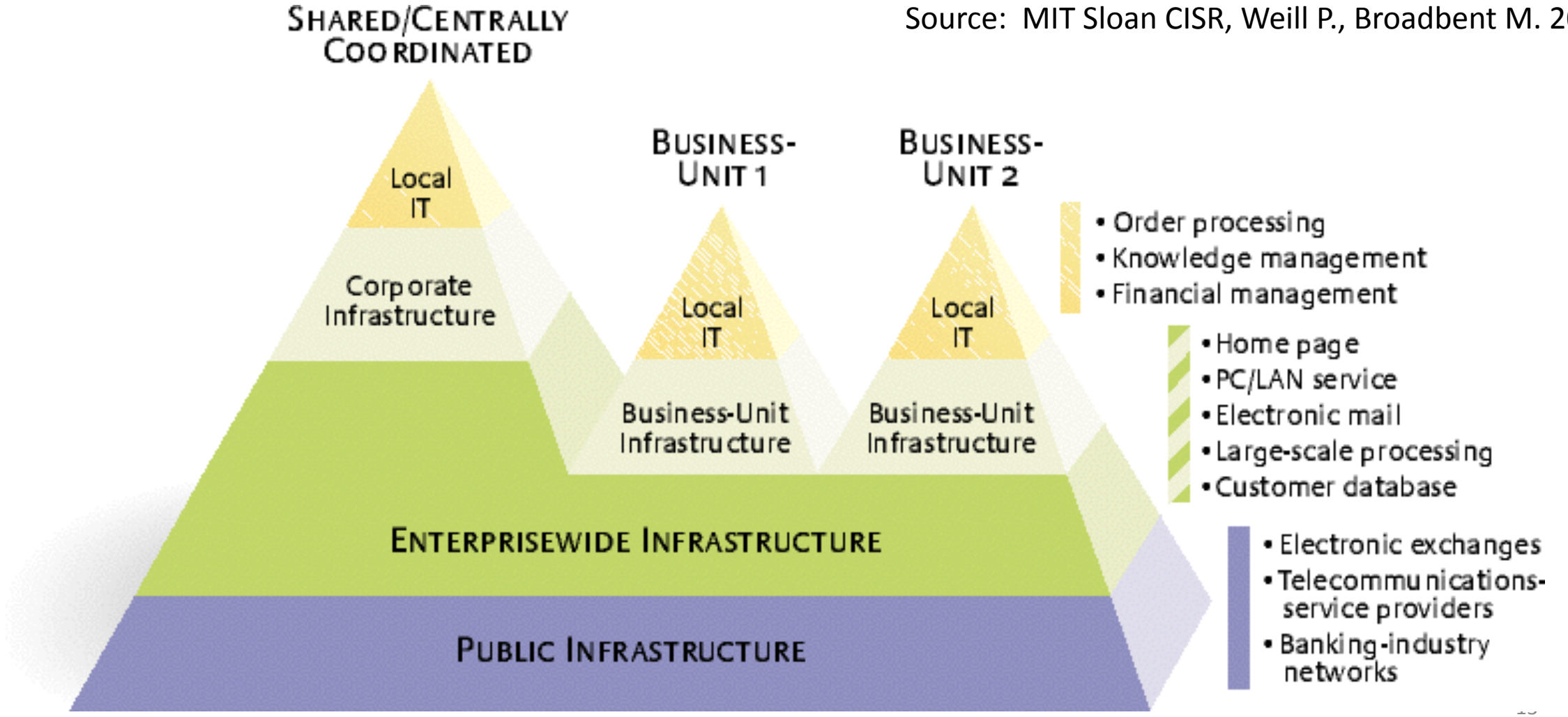
1994: Wentworth Research Program



Source: George Cox, Time to Reshape the IS Department? Wentworth Research Program (now part of Gartner EXP, Stamford, CT), June 1994.

Centre for Information Systems Research (CISR) Multi-unit Portfolio Model

Source: MIT Sloan CISR, Weill P., Broadbent M. 2002



What is a Service?

- Basic Definition
  - Inputs + Functionality = Output

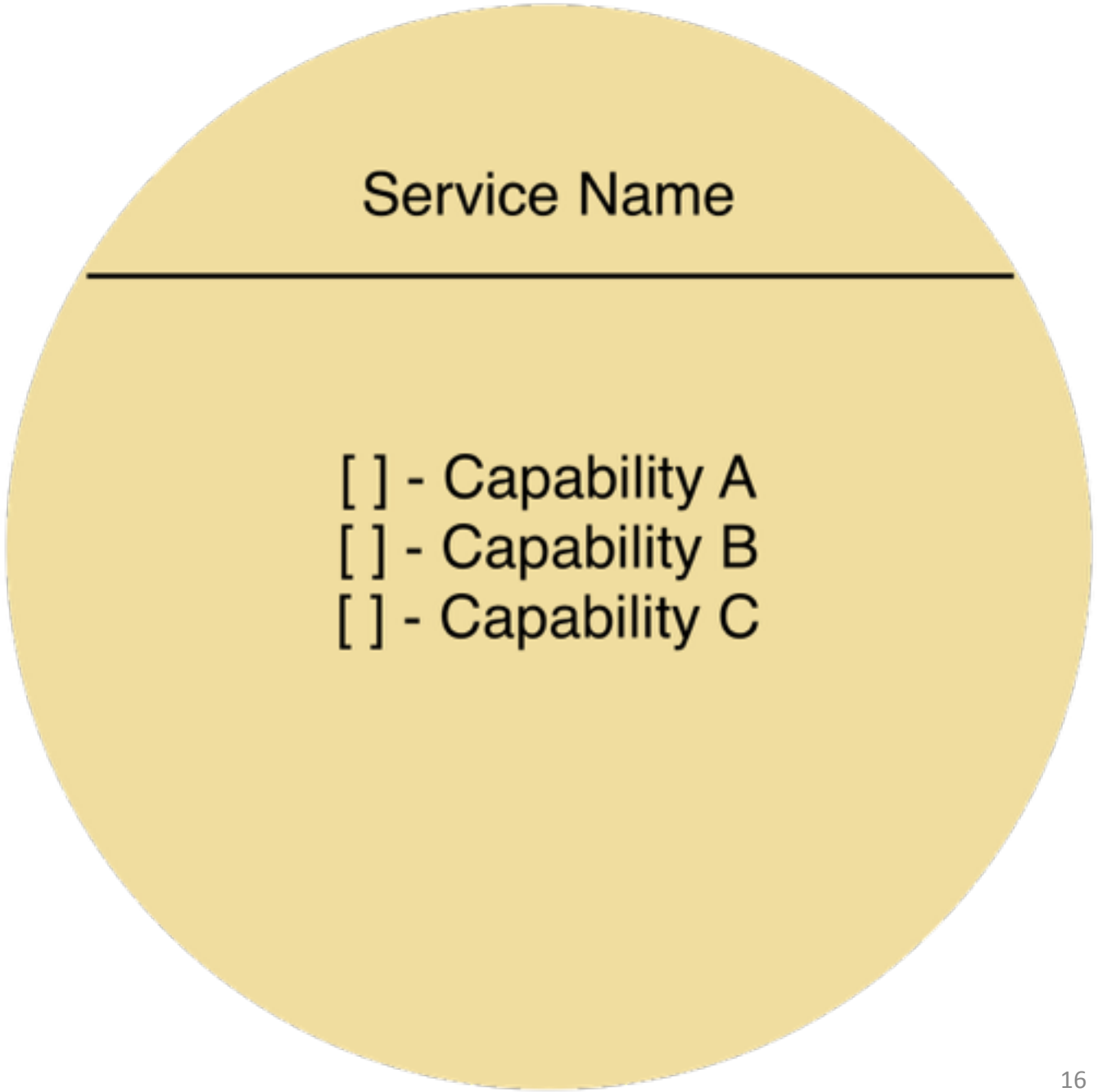
- Formal Definition
  - See: Journal of Software, July 2006
  - Aliaksei Yanchuk, Alexander Ivanyukovich, Maurizio Marchese “Towards a Mathematical Foundation for Service-Oriented Applications Design”

What is a Service?

- Basic Definition
  - Inputs + Functionality = Output

- Practical Definition
  - Inputs = (effort, data, contract, connection)
  - Functionality (unknown to user -> technology, process, people)
  - Output = (results)

What is a Service?



Source: http://servicetechbooks.com/pdf/SOA_Principles_Poster.pdf

Information Technology Infrastructure Library (ITIL)

Quotes from: http://en.wikipedia.org/wiki/ITIL

- **Service Strategy**
  - provides guidance on clarification and prioritization of service-provider investments in services.
- **Service Design**
  - provides good-practice guidance on the design of IT services, processes, and other aspects of the service management effort.
- **Service Transition**
  - relates to the delivery of services required by a business into live/ operational use, and often encompasses the ”project” side of IT rather than.

Information Technology Infrastructure Library (ITIL)

Quotes from: http://en.wikipedia.org/wiki/ITIL

- **Service Operation**
  - aims to provide leading practice for achieving the delivery of agreed levels of services both to end-users and the customers (where ”customers” refer to those individuals who pay for the service and negotiate the Service Level Agreements (SLAs).
- **Continual Service Improvement**
  - aims to align and realign IT services to changing business needs by identifying and implementing improvements to the IT services that support the business processes.

Enterprise Architecture (EA)

Gartner:

- Enterprise architecture (EA) is a discipline for proactively and holistically leading enterprise responses to disruptive forces by identifying and analyzing the execution of change toward desired business vision and outcomes. EA delivers value by presenting business and IT leaders with signature-ready recommendations for adjusting policies and projects to achieve target business outcomes that capitalize on relevant business disruptions. EA is used to steer decision making toward the evolution of the future state architecture.

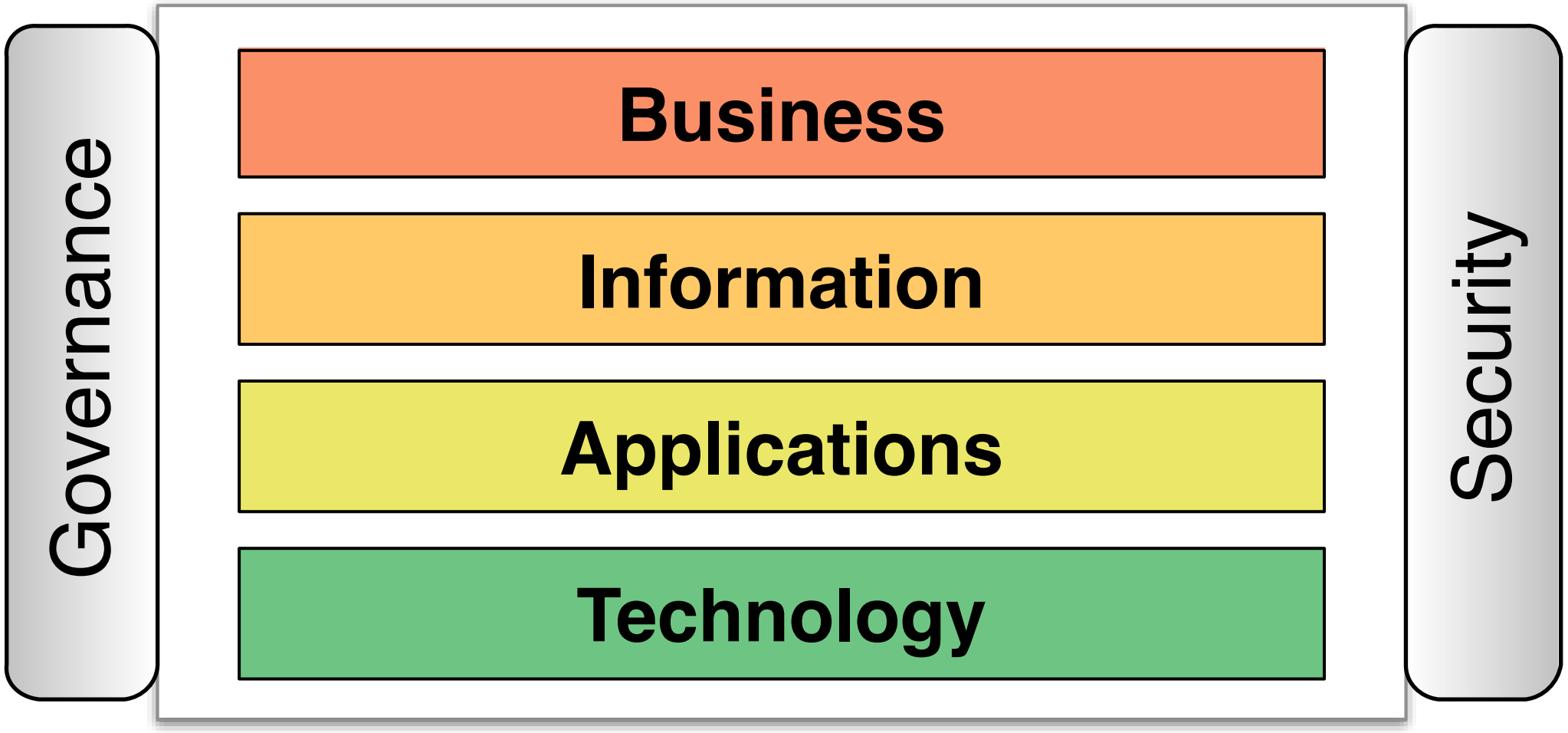
Source: http://www.gartner.com/it-glossary/enterprise-architecture-ea/

Enterprise Architecture (EA)

Human readable:

- Enterprise architecture (EA) is a discipline for taking a structured approach to studying, documenting, designing, planning, and facilitating change within an organization. The goal of EA is to allow an enterprise to better identify high-value opportunities and help it effectively capitalize on them.

Enterprise Architecture (EA)



Zachman Framework for Enterprise Architecture (EA)

Information Systems Management In Practice (7th Ed.) by McNurlin, C.B; Sprague, R.H. [Prentice Hall, 2008]

- Zachman Columns:
  - **"What"** Things + Data
  - **"How"** Processes
  - **"Where"** Network
  - **"Who"** People
  - **"When"** Events + Times
  - **"Why"** Strategies + Motivations
- Zachman Rows:
  - **"Contextual"** Planner /Enterprise View
  - **"Conceptual"** Owner / Business View
  - **"Logical"** Designer / Architect View
  - **"Physical"** Builder / Engineer View
  - **"Detailed"** Technician View
  - **"Functional"** Operator View

Zachman Framework for Enterprise Architecture (EA)

Information Systems Management In Practice (7th Ed.) by McNurlin, C.B; Sprague, R.H. [Prentice Hall, 2008]

	Why	**How**	**What**	**Who**	**Where**	**When**
Contextual (Enterprise)	Goal List	Process List	Material List	Organizational Unit & Role List	Geographical Locations List	Event List
Conceptual (Business)	Goal Relationship	Process Model	Entity Relationship Model	Organizational Unit & Role Relationship Model	Locations Model	Event Model
Logical (Architect)	Rules Diagram	Process Diagram	Data Model Diagram	Role Relationship Diagram	Locations Diagram	Event Diagram
Physical (Engineer)	Rulse Specification	Process Function Specification	Data Entity	Role Specification	Location Specification	Event Specification
Detailed (Technician)	Rules Details	Process Details	Data Details	Role Details	Location Details	Event Details

Enterprise Architecture Body of Knowledge (EA BOK)

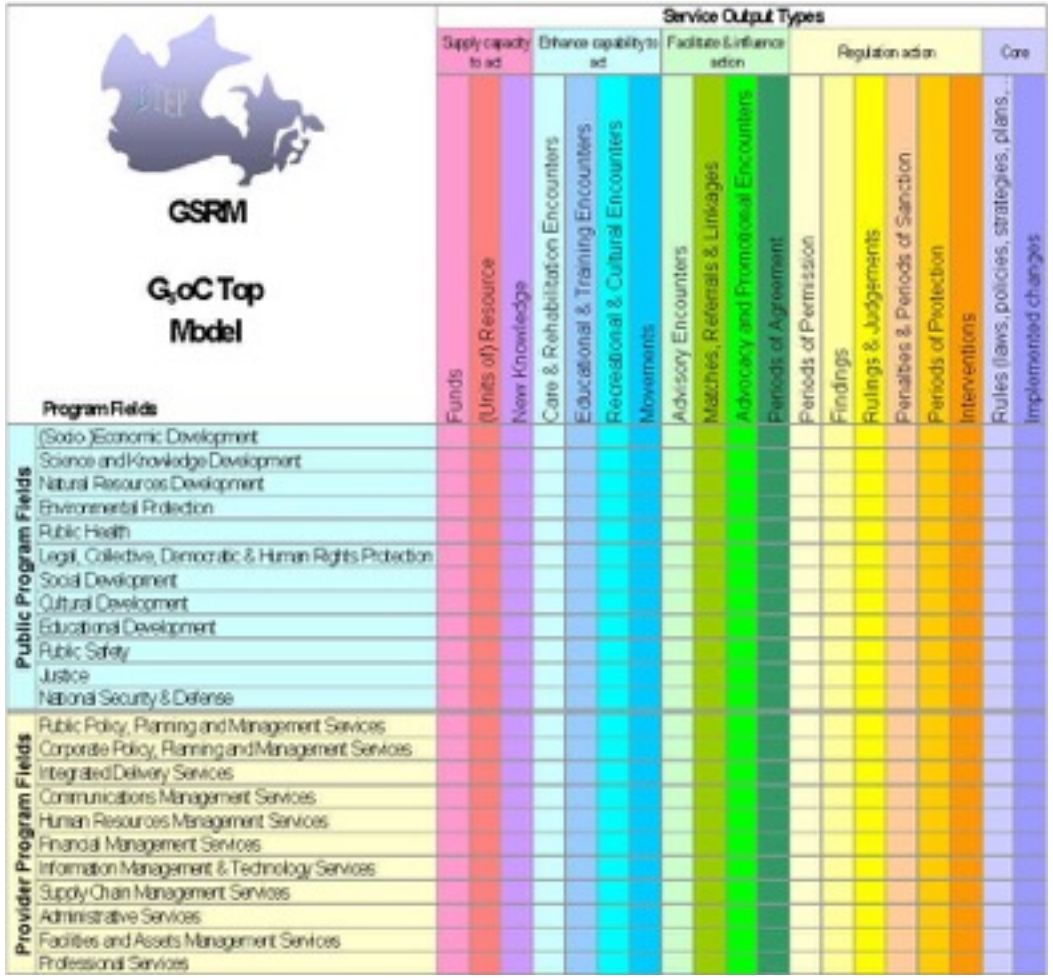
- Zachman is one of many Frameworks.
- Enterprise Architecture has many approaches, not one size fits all.

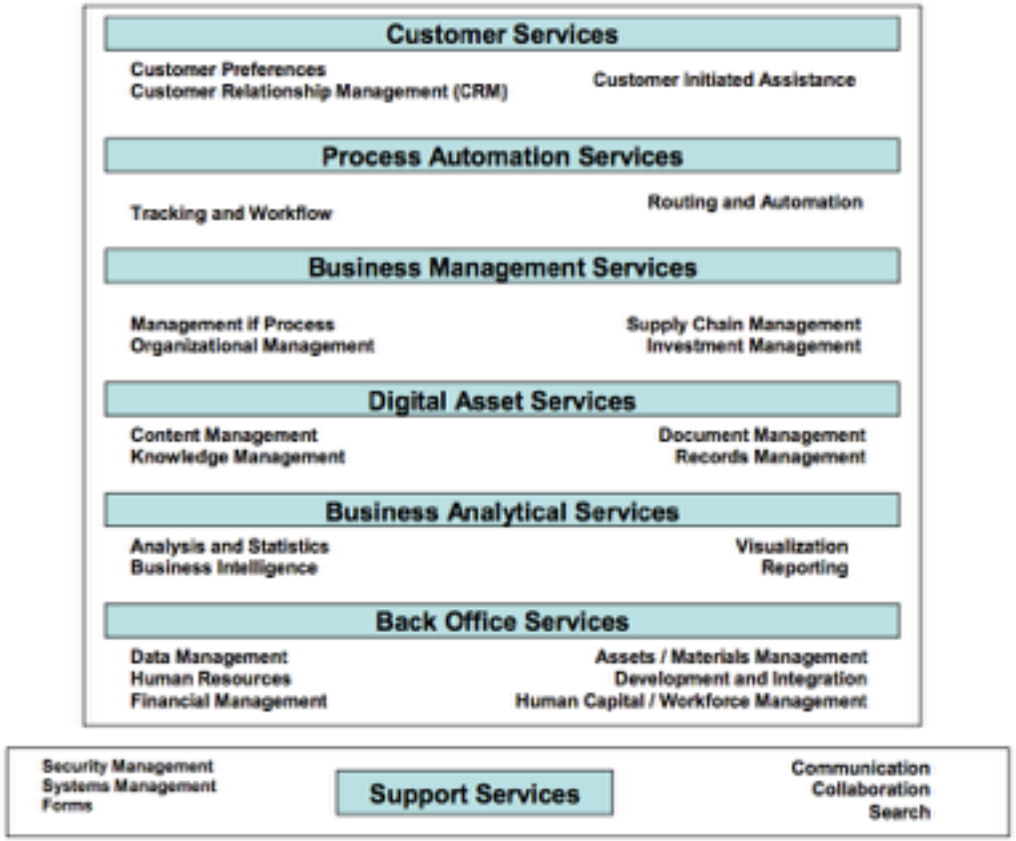
- To learn more visit the MITRE EA BOK:
  - http://www2.mitre.org/public/eabok/

Business Service Reference Models

- Governments of Canada Strategic Reference Model (GSRM)
  - Canadian Government Model, built up from Municipal Models.
  - Defines Service Types and Service Output Types
  - Supports Modelling Languages like Service Integration and Accountability (SIAM) and Program Service Alignment Model (PSAM)
- Office of Management and Budget (OMB) Federal Enterprise Architecture (FEA) Service Component Reference Model (SRM)
  - US Federal Government Model, built up from Agency Models.

Business Service Reference Models





BTEP GSRM

OMB FEA SRM

A Service Oriented Enterprise (SOE)

… is created when both Business and IT are Service-Oriented in their endeavours.

Alignment is far easier to govern when business and IT are modelled as connected value chains from back office through business to our clients.

Service-Oriented Architectures (SOA)

- In software engineering, a service-oriented architecture (SOA) is a set of principles and methodologies for designing and developing software in the form of interoperable services. These services are well-defined business functionalities that are built as software components (discrete pieces of code and/or data structures) that can be reused for different purposes. SOA design principles are used during the phases of systems development and integration.”

SOA Principles: Quick and Dirty

- loosely couple at all costs
- never require a particular operating system or technology
- keep services unassociated until runtime
- do not allow any embedded links between services
- only communicate over documented channels
- only communicate through documented interfaces
- to build on top of other services (compose) at quality and to
- spec, SLA underpinning contracts (UCs) are required

SOA Principles: Thomas Erl View

http://www.soaposters.com/

- **Standardized Service Contract**
  - Services within the same service inventory are in compliance with the same contract design standards.
- **Service Loose Coupling**
  - Service contracts impose low consumer coupling requirements and are themselves decoupled from their surrounding environment.
- **Service Abstraction**
  - Service contracts only contain essential information and information about services is limited to what is published in service contracts.

SOA Principles: Thomas Erl View

http://www.soaposters.com/

- **Service Reusability**
  - Services contain and express agnostic logic and can be positioned as reusable enterprise resources.
- **Service Autonomy**
  - Services exercise a high level of control over their underlying runtime execution environment.
- **Service Statelessness**
  - Services minimize resource consumption by deferring the management of state information when necessary.

Why SOA? Ask Stevey!

- Case: Amazon vs. Google
- Steve Yegge's "Stevey's Google Platforms Rant"
  - Engineer at Google released a rant on Google+ around Oct 2011.
  - A user error with Google+ led to a Google employee posting a rant against Google.
  - He had worked at Amazon before Google and ranted about where Google was failing.

Why SOA? Ask Stevey!

In 2002, Jeff Bezos (founder + CEO of Amazon) issued a mandate.

1. All teams will henceforth expose their data and functionality through service interfaces.
2. Teams must communicate with each other through these interfaces
3. There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.

Why SOA? Ask Stevey!

4. It doesn’t matter what technology they use. HTTP, Corba, Pubsub, custom protocols – doesn’t matter. Bezos doesn’t care.
5. All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.

Why SOA? Ask Stevey!

Lessons from a massive undertaking of building SOA at Amazon:

- pager escalation can get hard. need metrics and reporting
- every single one of your peer teams becomes a potential denial of service
- monitoring and QA are the same thing in SOAs
- a universal service registration mechanism is a powerful thing to have
- follow-on benefits are compelling

Why SOA? Ask Stevey!

- Steve then explains... as hard as SOA was, it was the Right Thing to do.
- He goes on to stress that Amazon's abilities as a provider of Infrastructure and a Platform far outstrip Google because of one ultimate thing:
  - Accessibility!
- If someone should be able to access something and cannot get it through a Service, it represents a HUGE roadblock to the Organization's success.

Why SOA? Ask Stevey!

Moral of the story:

- There is evidence that an organization is able to thrive in their market after adopting an SOA mandate. They were able to develop marketable value-add functionality following their adoption of SOA. They accomplished this by imposing a requirement that everyone always use ’Services.’ Amazon used a series of Lego blocks to combine functionality in a wide variety of ways.

Governance: the questions we need to ask.

1. If we continue duplicating high cost / low impact work across our organizations (province?) what is the Opportunity Cost?
2. How are resources provisioned to support activities? If funds flow to decentralized parties / agents, how do we convince them to share services?
3. Can a discipline like Enterprise Architecture help?
4. How can we help our business clients see the value of modelling their activities as services? How do we build complete models that allow the value chain from back office to client to be measured?

Game Plan for building an SOE

1. Control what you can.
   - Get IT in order, build radical Service-Orientation within your organizations.
2. Learn what it means to be Service-Oriented. Trial by fire.
   - Make your mistakes, pick yourself up, and learn from them.
   - Adopt LEAN and Agile mindsets. Business value is the primary driver. Fail early, learn often is the new IT mantra.
3. Help the Business learn to think radical Service-Orientation.

OUCC 2015

Inspiring Innovation

Questions & Answers