# Service-oriented Architecture and the Project Manager

## (or everything you always wanted to know about software reuse... but were afraid to ask)

**William Buckingham**
**Toronto 2010 February 27**

- **PMI-SOC**
- **GTISLIG**

# Agenda

- Introduction
- Software Reuse
- Service-oriented Architecture
- Game Plan
- Summary
- Q&A

# Intro: Goal

- Offer insight into how…
  - *service-oriented architecture* (SOA) can support the practice of *software reuse*
  - PMs can exploit *software reuse* and SOA to…
    - make long-term strategic investments
    - achieve short-term project savings
- Make sure you don't fall asleep

# Intro: Target Audience

- Managers of projects…
  - that provide reusable software
  - that consume reusable software
  - that don't fall into either category… but should
- Senior management and executives that want to benefit from software reuse and SOA

# Intro: Acronyms

| | |
|---|---|
| PMI | Project Management Institute |
| PMI-SOC | PMI Southern Ontario Chapter |
| GTISLIG | Greater Toronto Information Systems Local Interest Group |
| SOA | Service-oriented Architecture |

# Intro: The Speaker

- William Buckingham
  - Nominal founder of *Canadians Resisting Acronym Proliferation*
  - 30+ year veteran of the IT industry
    - One-time 0S/360 Assembler programmer
    - Decade as member of international council on mainframe computing
  - Works with a major Canadian FI
    - Fifteen years as IT Architect
    - Wrote organisation's first SOA paper in 1999
  - Contributor to the book *Secrets of SOA: An Enterprise View on Service-Oriented Architecture Deployment Revealed*

# Agenda

- Introduction
- Software Reuse
- Service-oriented Architecture
- Game Plan
- Summary
- Q&A

# Software Reuse: Successes

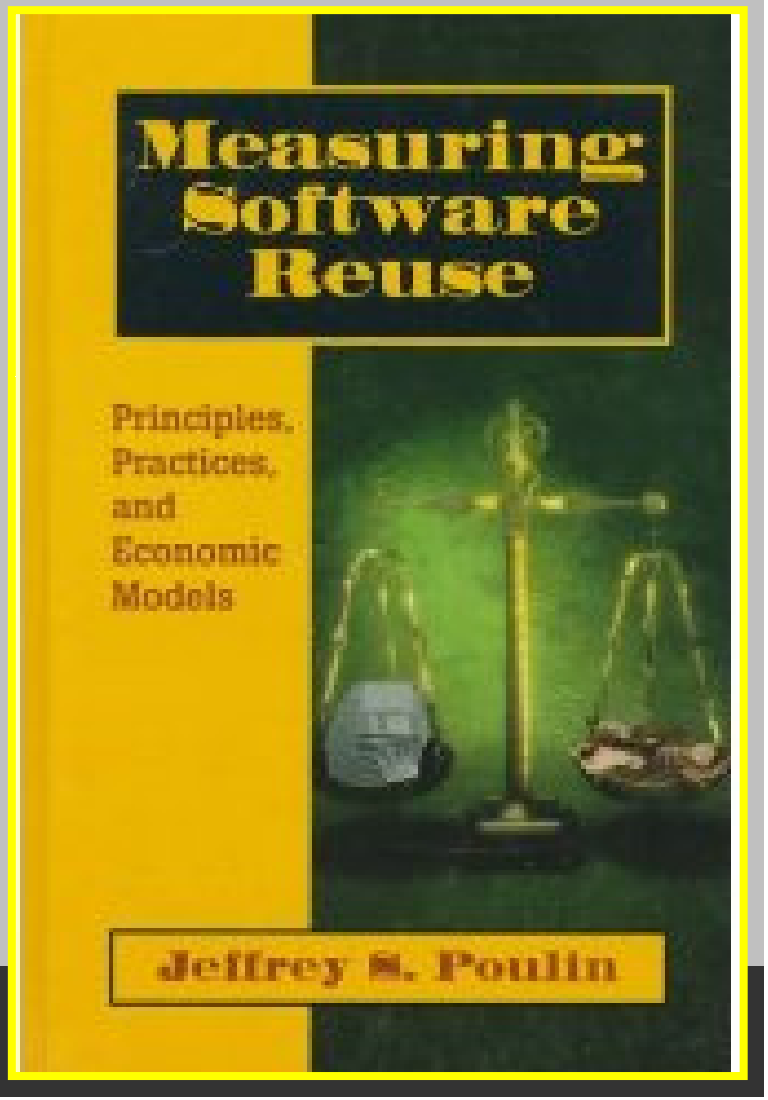
| | |
|---|---|
| **American Navy's *Restructured Naval Tactical Data Systems*** | **26% reduction in development and maintenance hours** |
| **Raytheon's Missile Systems Division** | **50% increase in productivity** |
| **Fujitsu's *Electronic Switching Systems*** | **From 20% *on schedule delivery* to 70%** |
| **US Army's *Tactical Command and Control System*** | **$479.9M in cost avoidance** |
| **Magnavox developed *Force Fusion System Prototype*** | **20% of projected, estimated time** |

# Software Reuse: Failures

- Documented causes
  - Organisational impediments
  - Economic impediments
  - Administrative impediments
  - Political impediments
  - Psychological impediments

# Software Reuse: Cost/Benefit

- From industry experts
  - **20%**: the expected costs of reusing instead of building
  - **60%**: the average investment to make an asset “reusable”
  - Both 20% and 60% are based on broad studies
    - Your mileage may vary



# Software Reuse: “Cost”

- Cost?
  - Is the cost of software reuse an expense?
  - My view?  It’s an investment!

# Software Reuse: “asset”

**as·set**
ˈ**æs ɛt/ [as-et]**
***–noun***
**1.**a useful and desirable thing or quality: ***Organizational ability is an asset.***
**2.**a single item of ownership having exchange value.
**3.assets,**
**a.**items of ownership convertible into cash; total resources of a person or business, as cash, notes and accounts receivable, securities, inventories, goodwill, fixtures, machinery, or real estate (**opposed to liabilities** ).
**b.*Accounting*.** the items detailed on a balance sheet, esp. in relation to liabilities and capital.
**c.**all property available for the payment of debts, esp. of a bankrupt or insolvent firm or person.
**d.*Law*.** property in the hands of an heir, executor, or administrator, that is sufficient to pay the debts or legacies of a deceased person.
**Origin:**
1525–35; back formation from ***assets,*** in phrase ***have assets,*** lit., have enough (to pay obligations) < AF, OF ***asez*** enough. **See assai 1**

***Related forms:***

**as·set·less, *adjective***

(from Dictionary.com http://dictionary.reference.com/browse/asset)

# Software Reuse: “asset”

- All assets have value
  - Development work months
  - Acquisition cost
  - All project costs
  - Integration, testing, deployment
  - … even something based on Open Source

# Software Reuse: “asset”

- Calculating that value
  - Try to be as inclusive as possible
  - Try to be as accurate as possible
  - Keep it simple
  - Art versus science
  - No precise formula

# Agenda

- Introduction
- Software Reuse
- Service-oriented Architecture
- Game Plan
- Summary
- Q&A

# SOA: Definition

- "… a **service-oriented architecture** (**SOA**) is a flexible set of design principles used during the phases of systems development and integration. A deployed SOA-based architecture will provide a loosely-integrated suite of services that can be used within multiple business domains."
  - Wikipedia

# SOA: Definition

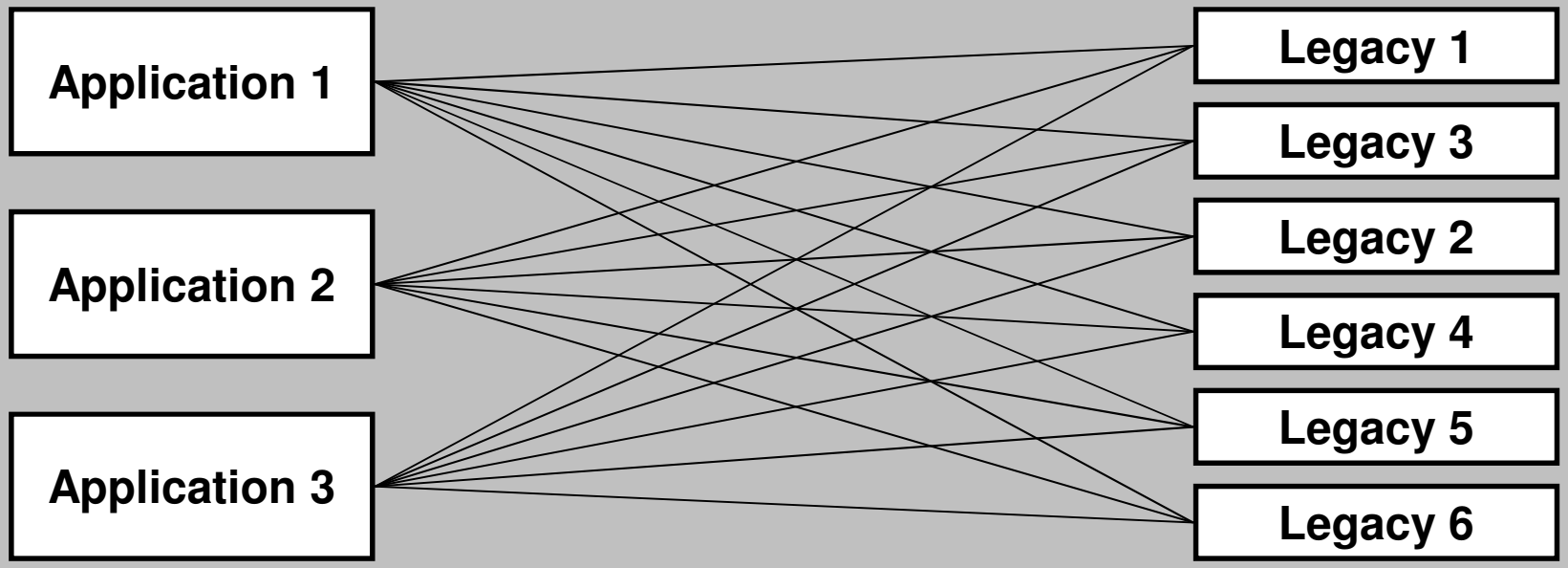
- "Service Oriented Architecture (SOA) is a business-centric IT architectural approach that supports integrating your business as linked, repeatable business tasks, or services."
  - IBM

# SOA: Definition

- Service-oriented Architecture
  - Hyphenated

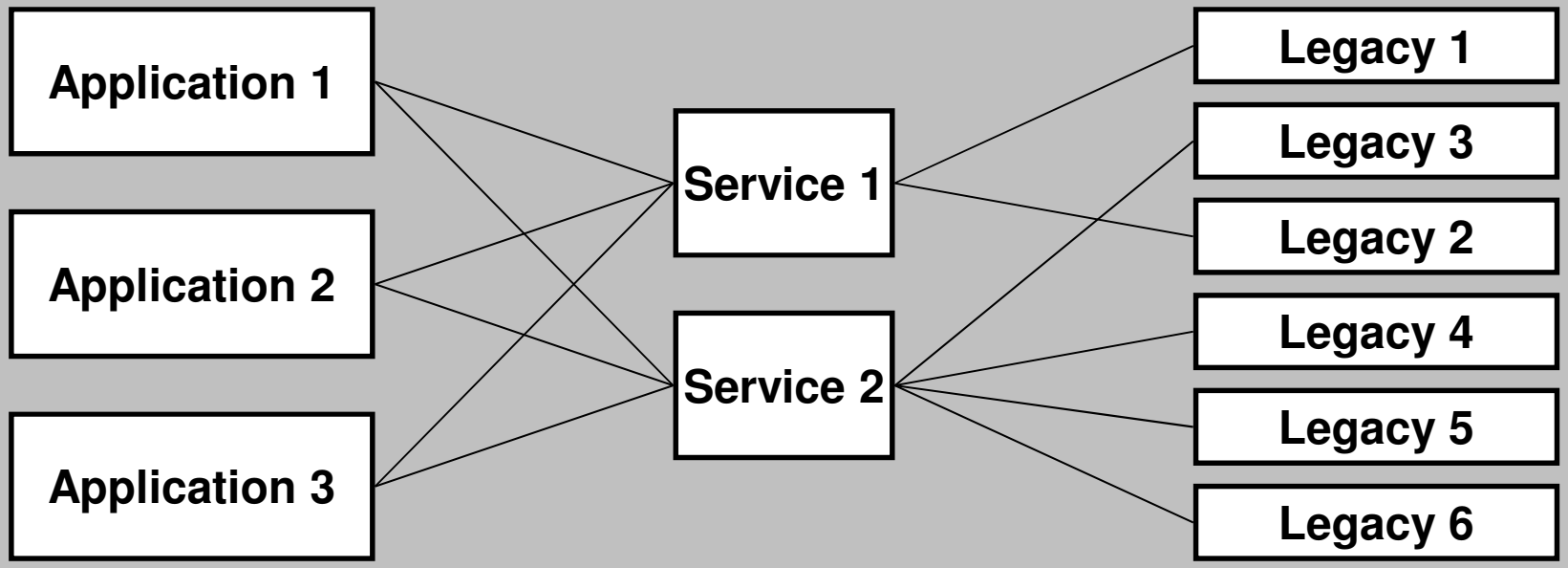
# SOA: Basic architecture

- Without SOA
  - Applications x Legacy Systems = 18 Connections



# SOA: Basic architecture

- With SOA
  - (Applications x Services) + (Services x Legacy Systems) = 10 Connections



# SOA: Definition

- The What: verbs and nouns
  - Assets in SOA can express the business in terms non-IT staff should recognise
  - Nouns
    - “Account”, “Client”, “Location”…
    - Objects in old object-oriented (OO) terms
  - Verbs
    - “New Client”, “Sale”, “Credit Check”…
    - Business Processes

# SOA: Definition

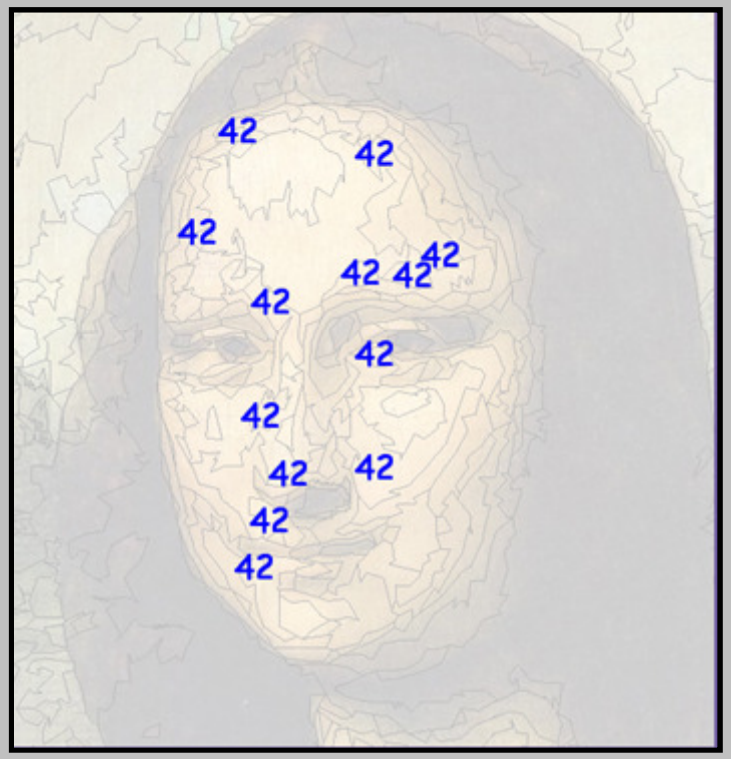
- SOA a great opportunity to…
  - create new relationship with the business
  - express the “Business Architecture”
- Manage the scope
  - Only one business domain
  - Avoid “Enterprise SOA”
  - Mixing lines-of-business causes problems
  - Account, account and account!

# SOA: Definition

- Service-oriented Architecture
  - It is architecture
  - Would you build a house without an architecture?
  - Would you build your business without an architecture?

# SOA: How To

- Paint-by-numbers
  - Architecture: The Big Picture
    - The lines and canvas
    - The idea and structure
  - Projects: Just-in-time
    - The paint and brushes
    - The resources
  - Result: A work of art
    - Poetic license…!



# SOA: Benefits

- Service-oriented Architecture
  - Reduction in maintenance costs may be 2 to 10 times more
  - Improvements in quality through more concentrated QA effort
  - Time-to-market
  - Reuse savings may be even greater than with basic software reuse!

# SOA: What it isn't!

- SOA is not product
  - No vendor can sell you "an SOA"
  - An SOA should be a representation of the business
    - Your business!
    - Industry models might accelerate the process but inevitably require customisation and extension

# SOA: What it isn't!

- SOA is not a specific technology
  - No vendor can sell you "an SOA"
  - Many technologies can support an SOA
    - Warning: each can also be misused and run poorly-architected non-reusable applications
  - Other technologies can also support SOA

# SOA: What it isn't!

- SOA is not…
  - Business Process Management (BPM)
  - Web Services
  - Enterprise Service Bus (ESB)… is not *Everybody's Silver Bullet!*

# SOA: What it isn't!

- SOA: Service-oriented Anarchy
- SOC: Service-oriented Chaos
- JBOS: Just a Bunch of Services

# SOA: A comparison

| | Without Reuse | With Reuse | With SOA |
|---|---|---|---|
| **Assets used by more than one application** | No | Yes | Yes |
| **Encapsulation of legacy** | No | Maybe | Yes |
| **Improved data quality** | No | Maybe | Yes |
| **Consistent interface** | No | No | Yes |
| **Reduction in maintenance costs may be 2 to 10 times more** | No | Yes | Yes |
| **Improvements in quality through more concentrated QA effort** | No | Yes | Yes |
| **Time-to-market** | No | Yes | Yes |

# Agenda

- Introduction
- Software Reuse
- Service-oriented Architecture
- Game Plan
- Summary
- Q&A

# Game Plan: Executives

- Support the development of a Service-oriented Architecture
  - Architecture
  - Supporting technologies
  - Executive buy-in critical – short-term and long-term
- Funding
  - Fund the development of reusable assets that conform to SOA
  - Favour projects that consume reusable assets
  - Consider PPM methodologies

# Game Plan: Executives

- Communicate
  - The architecture
  - The process
- Set targets for each manager or area
  - Number of instances of reuse
  - Based on asset value
  - Incentives!
- Govern

# Game Plan: Managers

- Commit to development of software defined by the SOA
  - Budget to allow for the development of software defined by the SOA
  - Think investment!
  - Opportunity to show commitment to enterprise direction
- Commit to use of software defined by the SOA in other development
  - Opportunity to save time and money on your project

# Game Plan: Managers

- Manage
  - Communicate to staff the value of looking for opportunities to reuse
  - Work towards targets
  - Walk the talk

# Agenda

- Introduction
- Software Reuse
- Service-oriented Architecture
- Game Plan
- Summary
- Q&A

# Summary

- Software reuse is good
- Software reuse in the context of SOA is better
- Cultural change
- Targets
- Think “investment” not “expense”

# Summary

- Have reuse stories?
  - Victories?
  - Scars?
  - Let's talk!  wgcb@sympatico.ca


Questions
&
Answers