NPS-AM-10-055

EXCERPT FROM THE

PROCEEDINGS

OF THE

SEVENTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM THURSDAY SESSIONS VOLUME II

Acquisition Research

Creating Synergy for Informed Change

May 12 - 13, 2010

Published: 30 April 2010

Prepared for: Naval Postgraduate School, Monterey, California 93943

The research presented at the symposium was supported by the Acquisition Chair of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request Defense Acquisition Research or to become a research sponsor, please contact:

NPS Acquisition Research Program
Attn: James B. Greene, RADM, USN, (Ret.)
Acquisition Chair
Graduate School of Business and Public Policy
Naval Postgraduate School
555 Dyer Road, Room 332
Monterey, CA 93943-5103
Tel: (831) 656-2092
Fax: (831) 656-2253
E-mail: jbgreene@nps.edu

Copies of the Acquisition Sponsored Research Reports may be printed from our website www.acquisitionresearch.net

Meaningful Cost-Benefit Analysis for Service-Oriented Architecture Projects

Lloyd Brodsky—Lloyd Brodsky is a solution architect senior consultant in the Federal Consulting Practice of Computer Sciences Corporation. He frequently consults on governance and acquisition issues in service-oriented architecture for a variety of government clients. He holds a doctorate in management information systems from the Sloan School of the Massachusetts Institute of Technology and an MBA in finance from the business school of the University of California, Berkeley. He also holds the PMP and CISSP certifications and completed COTR training while employed at the Department of Veterans Affairs.

Abstract

This paper argues that proper cost-benefit analysis of service-oriented architecture projects is not possible without explicit identification of SOA-specific tasks in the work breakdown structure (WBS), so that those costs are explicitly estimated in the budget, are explicitly in the integrated master schedule, and appear on earned value and other reports. It deconstructs the traditional stories for financially justifying SOA and identifies SOA-specific activities that should be added to WBSs to enable tracking of costs and schedules. It also identifies specific research questions that can only be answered with data gathered through such task-level cost accounting.

Introduction

The central point of this paper is quite simple: You cannot do proper return on investment (ROI) or earned value analysis (EVA) of a service-oriented architecture (SOA) project without a work breakdown structure that explicitly identifies SOA-specific activities in a way that facilitates comparing investments and benefits. This often does not happen because SOA has additional steps in development and operation that either do not exist or are less important in traditional development. This paper deconstructs the traditional arguments for SOA to identify these activities and shows how those costs come to be commingled with other development and maintenance activities. The paper argues that this comingling impairs the ability to diagnose problems and recommend best practices. It recommends that acquisition professionals require inclusion of important SOA tasks in the work breakdown structure and the integrated master schedule, as well as requiring reporting on a range of SOA cost issues.

Service-Oriented Architecture Review

SOA is a term with multiple meanings. It refers to both an architectural style and to web services technology. The architectural style calls for dividing needed functionality into separately built business function-related services. Applications are composed of such loosely coupled services, which usually communicate with each other via a web-service interface. The underlying idea is that it would be cheaper and faster to build or modify applications by composing them out of limited-purpose components that can communicate with each other because the components strictly adhere to interface rules. There are three standard arguments for why this would result in financial savings and be a positive net present value deal (+NPV):

- **Interfaces:** SOA replaces m*n point-to-point interfaces with m+n service interfaces. In exchange for an upfront investment in defining a common vocabulary and common interfaces and a continuing cost in governance and ESB maintenance, this would save maintenance costs over time. The savings would come from:
  - Having fewer interfaces to maintain
  - Reduced "information archeology" costs when making changes due to the tighter configuration management needed to get services to work
  - Reduced certification and accreditation costs due to fewer and better documented interfaces and services
  - The hiding of connection details by the enterprise service bus, if one were used[1].
- **Authoritative databases:** SOA stores data once and publishes it as a service for all the applications that need it. This reduces costs by:
  - Only storing one version and,
  - Indirectly through separating application code from the data and,
  - Indirectly through better data quality – if you have multiple versions either they are all the same or something is wrong.
- **Reuse:** Money is saved to the extent that new applications and changes are accomplished through reusing existing services. An example would be reusing a security service in all applications that required authentication before use.

Cost Accounting Issues

As managers, we would naturally be interested in projecting these costs when planning and in gathering actual data in production to see how accurate those estimates were. There are also a number of more narrow SOA cost-accounting issues we would like to drill down to. These include:

- **Governance overhead:** A central part of SOA doctrine is the absolute requirement of a centralized governance organization that creates local standards and enforces both those and other applicable standards to ensure interoperability and architectural conformance. This review layer adds cost and time, but what the appropriate share of budget should be is not well understood.
- **Vocabulary synchronization cost:** It is also SOA doctrine that legacy systems can be made available as services by building an interface layer that maps the existing vocabulary to the ontology. It is quite possible than a "market analysis" of the demand for potentially-sharable information could lead to savings from not sharing information that is in little demand.

[1] Applications that read from flat files need to be changed if the record size or layout changes, whether that change affects the program or not.

- **Timing of ESB and registry installation:** Enterprise service busses and service registries play a valuable role in masking connection complexity and in making developed services findable. However, due to rapid technological evolution of COTS offerings, there is some question as to the value of introducing their use before a critical mass of available services is achieved.
- **Certification & accreditation relationship:** The standardization and tighter configuration management associated with SOA should drive down C&A costs.
- **Software best practices support services:** It is tacitly assumed that web-services interfaces are both well documented and stable, so that third parties can successfully use them by following the instructions. While good documentation and configuration is considered a best practice whether doing SOA or not, the possible damage is greater in SOA deployments.

The Work Breakdown Structure Problem

A work breakdown structure (WBS) is a hierarchical decomposition (tree structure) of the work required to accomplish a goal. It should be developed by starting with the end objective and successively redividing it into manageable components in terms of size, duration, and responsibility. However, it is often done as a modification of an existing WBS for a similar project. It is a required part of the DoD acquisition process.

The WBS is an essential starting input to both estimation and to scheduling. In essence, it provides the chart of accounts for a project. To get to a schedule, each task will have a duration, labor hours, and predecessor and successor tasks assigned to it. Pricing that labor gives a budget, not including materials. Each task in the WBS has hours, labor in labor categories and a duration. All earned value reports, all scheduling and all progress payments are keyed to the WBS. All work has to belong to and be "billed" to one of the tasks defined in the WBS. It follows that if you want to know what something costs, it needs to exist as a task in the WBS.

In large projects, the WBS is quite complex and can be as much as five or six levels deep. Usually, items at the same level of hierarchy are in the order they are executed, although this is not required. Traditionally, definition of the WBS is left to vendors with the integrated master schedule and price proposal based upon it included as part of the RFP response. More often than not, the organization of the WBS in software development follows the traditional "waterfall" method of system development. The primary constraint is that the WBS fulfills the requirements of the statement of work. In the Defense context, the foundation for WBSs is in *DoD Directive 5000.1* and in *MIL-HBDK-8881A*. The latter became a military standard on 1/9/2009. The Project Management Body of Knowledge published by the Project Management Institute (the basis for the PMP exam) also emphasizes the importance of the WBS in project management.

Since the development of the software WITHIN services is about the same as traditional development, we suggest that the distinctive feature of SOA from a WBS perspective is the tasks associated with developing the interfaces BETWEEN individual services. Unfortunately the practice of using project managers without a background in enterprise architecture has led to the development of WBSs that look more like traditional "waterfall" development, which leaves implicit the governance and common interoperability

infrastructure, which is independent of any single service's development and support team. It follows that the most helpful approach is to:

- **Divide the work into interfaces and services:** In SOA development there is a whole series of activities, such as ontology and interface development, whose function is to enable communication between services. These SOA-specific activities should be separately identified in the WBS.
- **Explicitly account for non-SOA activities that are critical for SOA:** While such activities as configuration management and technical documentation are not SOA activities per se, they are so important to SOA success they should be separately trackable as well.
- **See that the operations WBS is consistent with the development WBS:** Operations is often done by a different contractor and/or a different solicitation. If the idea is that, say, investment in web-service interfaces in development pays off in reduced cost of maintenance and change later, it would be helpful if the two WBSs facilitated that comparison.
- **Define relevant reports:** Because the value chains implied by the SOA benefit stories are fairly complex, some creativity is needed to define meaningful reports that aggregate associated things. For example, the cost of making changes to a service pursuant to a change order will involve governance review, coding, testing, independent validation, and certification and accreditation. In a mature SOA environment, this will involve not only aggregating managerial across different services maintenance organizations; it is also likely to involve different contractors. Understanding the complete cost impact and aggregating the information may be a challenge.

While there are not hard and fast rules for constructing a WBS in information technology development, the most common approach is to have tasks at the same level of hierarchy appear in order of start. Thus, in a number of WBSs examined anecdotally for this paper, at the top level they started with Enterprise Architecture and ended with Post-Deployment, with Development having the deepest structure and the largest number of leaf nodes.

Enterprise Architecture and Development are the two top-level activities most affected by SOA. To be consistent with the separation suggested above, this paper suggests the following new activities:

- **Enterprise architecture:** Enterprise architecture (EA) includes a diffuse range of engineering planning activities, which are bunched at the beginning of development but continue throughout development and into operations. EA is responsible for governance—the establishment of standards and subsequent review for compliance, which is essential to SOA success. Specific SOA tasks that would fall under governance involve:
  - **Planning**
    - **Ontology:** the development of controlled vocabularies for data interchange.
    - **Interface standards:** The standards that XML schemas and other technical artifacts are to follow. These are needed to

implement standardized interfaces and their associated management and error correction protocols.

- **Configuration management practices:** If future development and maintenance are to be able to connect to a service based solely on the technical documentation, configuration management and change management practices need to be very well controlled. EA needs to review and publish the definition of what a service is supposed to do and the service level agreement it is expected to follow.

- **Development review**
  - **Governance review: The governance entity will need to review the interoperability artifacts and service contracts produced by the development team.**
- **Run-time review and enforcement:** The governance entity needs to review the monitoring of service-level agreements and take appropriate corrective action in case of violations. Governance should be added. Also, configuration management should be increased.

- **Development:** By far the largest time and cost in any project is actually doing the development. This is typically subdivided into:
  - **Requirements and Design:** Specification and design the technical artifacts to implement planned interfaces. In addition, if a service is to be done by wrappering an existing capability, the vocabulary in that legacy system needs to be mapped into the ontology standard.
  - **Code and Unit test:** Implements the SOA-specific activities designed above.
  - **Integration test:** Independent verification and validation.

Conclusion

There is an old saying that you cannot manage what you cannot measure. By increasing the number of "moving pieces" in IT solutions, SOA increases the number of pieces that require measurement. Given the relative immaturity of the SOA paradigm, it is particularly important now, when best practices have not yet been established and the understanding of cause and effect is limited. Indeed, the inability to collect cost and schedule data at the task level may be part of the reason why so many case studies in SOA only present project-level estimates of averted cost.

It is well within existing authority for acquisition to require the WBS to make explicit anything that acquisition and the PMO would like to monitor. This will, in turn, assure that those SOA tasks appear explicitly in the integrated master schedule and on EVM reports. It is also possible for acquisition to standardize the terminology of WBSs across contracts in the same program, which could help assure tying investments made in development to their hoped-for payoff in operation. We recommend this happen and that acquisition also require reporting on SOA cost-benefit

THIS PAGE INTENTIONALLY LEFT BLANK

2003 - 2010 Sponsored Research Topics

Acquisition Management

- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- BCA: Contractor vs. Organic Growth
- Defense Industry Consolidation
- EU-US Defense Industrial Relationships
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Managing the Services Supply Chain
- MOSA Contracting Implications
- Portfolio Optimization via KVA + RO
- Private Military Sector
- Software Requirements for OA
- Spiral Development
- Strategy for Defense Acquisition Research
- The Software, Hardware Asset Reuse Enterprise (SHARE) repository

Contract Management

- Commodity Sourcing Strategies
- Contracting Government Procurement Functions
- Contractors in 21st-century Combat Zone
- Joint Contingency Contracting
- Model for Optimizing Contingency Contracting, Planning and Execution
- Navy Contract Writing Guide
- Past Performance in Source Selection
- Strategic Contingency Contracting
- Transforming DoD Contract Closeout
- USAF Energy Savings Performance Contracts
- USAF IT Commodity Council
- USMC Contingency Contracting

Financial Management

- Acquisitions via Leasing: MPS case
- Budget Scoring
- Budgeting for Capabilities-based Planning

- Capital Budgeting for the DoD
- Energy Saving Contracts/DoD Mobile Assets
- Financing DoD Budget via PPPs
- Lessons from Private Sector Capital Budgeting for DoD Acquisition Budgeting Reform
- PPPs and Government Financing
- ROI of Information Warfare Systems
- Special Termination Liability in MDAPs
- Strategic Sourcing
- Transaction Cost Economics (TCE) to Improve Cost Estimates

Human Resources

- Indefinite Reenlistment
- Individual Augmentation
- Learning Management Systems
- Moral Conduct Waivers and First-tem Attrition
- Retention
- The Navy's Selective Reenlistment Bonus (SRB) Management System
- Tuition Assistance

Logistics Management

- Analysis of LAV Depot Maintenance
- Army LOG MOD
- ASDS Product Support Analysis
- Cold-chain Logistics
- Contractors Supporting Military Operations
- Diffusion/Variability on Vendor Performance Evaluation
- Evolutionary Acquisition
- Lean Six Sigma to Reduce Costs and Improve Readiness
- Naval Aviation Maintenance and Process Improvement (2)
- Optimizing CIWS Lifecycle Support (LCS)
- Outsourcing the Pearl Harbor MK-48 Intermediate Maintenance Activity
- Pallet Management System
- PBL (4)
- Privatization-NOSL/NAWCI
- RFID (6)

- Risk Analysis for Performance-based Logistics
- R-TOC AEGIS Microwave Power Tubes
- Sense-and-Respond Logistics Network
- Strategic Sourcing

Program Management

- Building Collaborative Capacity
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Collaborative IT Tools Leveraging Competence
- Contractor vs. Organic Support
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to AEGIS and SSDS
- Managing the Service Supply Chain
- Measuring Uncertainty in Earned Value
- Organizational Modeling and Simulation
- Public-Private Partnership
- Terminating Your Own Program
- Utilizing Collaborative and Three-dimensional Imaging Technology

A complete listing and electronic copies of published research are available on our website: www.acquisitionresearch.org

THIS PAGE INTENTIONALLY LEFT BLANK

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CALIFORNIA 93943

www.acquisitionresearch.org