ISMA 2009 Conference
Chicago, Illinois

September 15, 2009

Finding the ROI in SOA

Terry Vogt

Booz Allen Hamilton

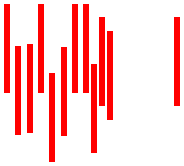
8283 Greensboro Drive

McLean VA 22102

Evaluating the ROI of SOA in Software From a Functional Perspective

Agenda

- SOA Concepts
- SOA Software
- Function Point Analysis for SOA
- Major Factors in ROI for SOA
- ROI Issues for SOA



SOA Concepts

- Service Oriented Architecture (SOA)
  - Provides a design framework for rapid and low-cost system delivery with improved total system-quality.
  - SOA infrastructure allows different components to exchange data with one another as they participate in business processes
- SOA is a methodology
  - SOA separates functions into distinct units, called “services”, which developers connect over a network allowing users to combine and reuse them in the production of business applications.
  - These services communicate with each other by passing data from one service to another, or by coordinating an activity between two or more services.

SOA Software

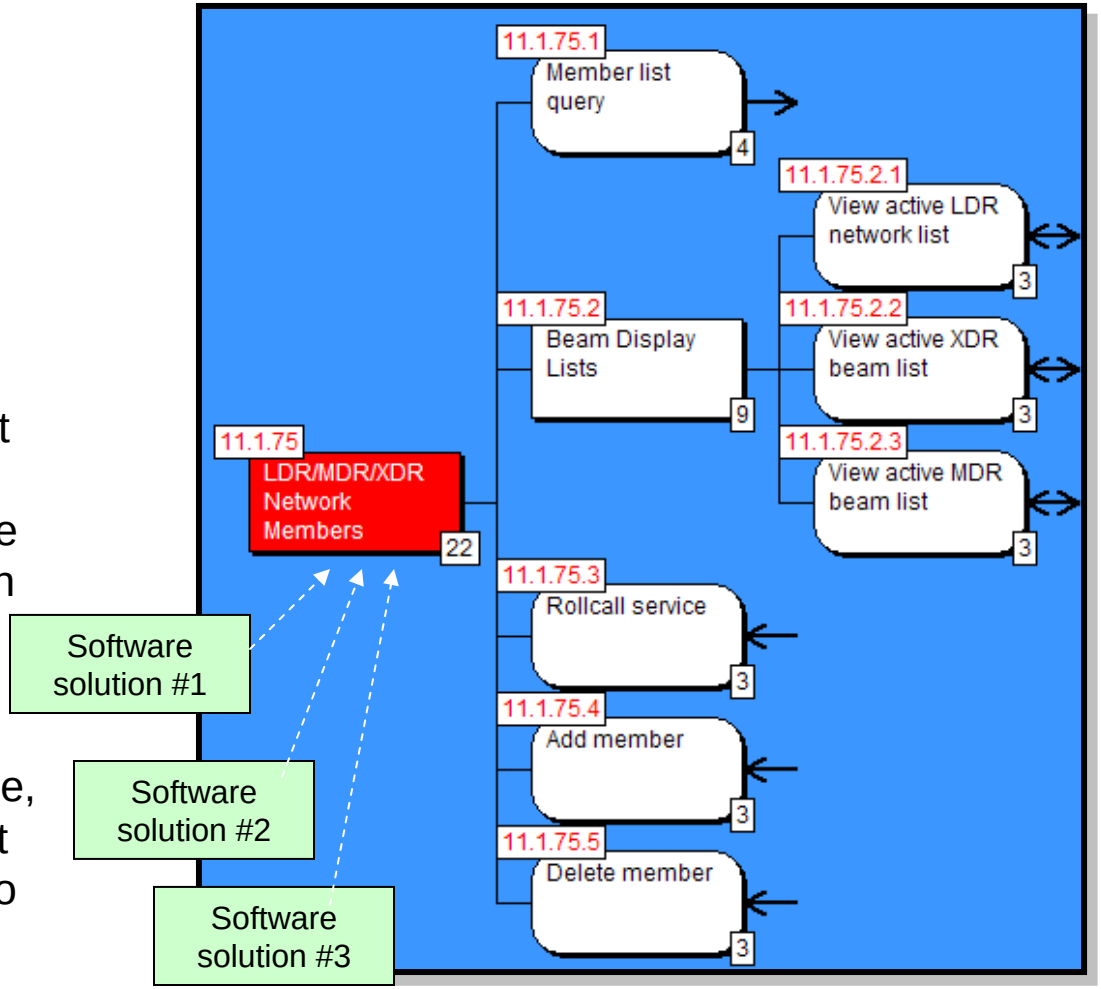
- Traditional Architecture Characteristics
  - Users access application functionality in stove piped systems
  - Systems implemented with point-to-point interfaces using multiple protocols – adds to complexity, cost, potential failure
  - Inflexible architecture – difficult to expand, adapt, evolve
- SOA Characteristics
  - Functional solutions regardless of technology used or source of solution
  - Flexible architecture – various standards applied but no absolutes in practice
  - Loosely coupled interfaces provide pathway to link functionality
  - Directory services identify components to link together and produce complete functional solutions
  - Rapid production of functional solution

Function Point Analysis for SOA

- Application requirements express the functionality required of an application. Function Point Analysis (FPA) is an approach to requirements-based sizing as input to cost estimation and cost alternative comparisons.
- FPA sizes applications in terms of the functionality (service) they provide, without regard for the means of implementation (coding language, platform, source, etc.)
- FPA can be applied in SOA at the component level, as well as at the service and application level, to model development and acquisition of software and to model integration development projects focused on linkage of services into applications
- FPA of SOA software:
  - Retains focus on requirements analysis
  - Requires business and technical knowledge
  - Measures size in a way that easily supports analysis of alternative solutions
  - Supports software reuse – Highlights identification of patterns of functionality

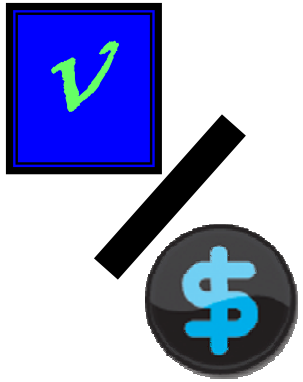
The direct connection between requirements and size provide a distinct benefit to using function points in SOA

- Systems requirements specification (SRS) and similar documents can be translated into function points
- Software components that match requirements can also be sized – this will include the size of any unused functionality present in the component
- Component software “Best fits” can be identified and costs of alternatives can be compared against normalized requirements size
- As requirements change over life cycle, their impact to size (and resulting cost and schedule) can be linked directly to those changes



ROI – Return on Investment

- ROI is typically defined as dollars obtained as a result of dollars invested
- Investment (cost) is relatively easy to determine. The “return” on many software projects can not be effectively determined or expressed in dollars obtained
  - The linkage between specific software or functionality and dollars returned is often obscure or entirely absent even where there is a dollar value associated with a business process supported by software
  - The benefits obtained through functionality implemented via software are often entirely non-financial
- The result is more properly expressed as “benefits” or “value” rather than “return”
  - Benefits may include faster response, ease of use, flexibility of user interaction, more options, etc.
  - Value may encompass improved customer satisfaction and retention, parity with competitors, and many other aspects



- There are methods of systematically capturing and quantifying the value of non-financial benefits
  - e.g. Balanced Scorecard approach, Value Measurement Methodology, others

ROI in Component Based Software – Investment (Cost)

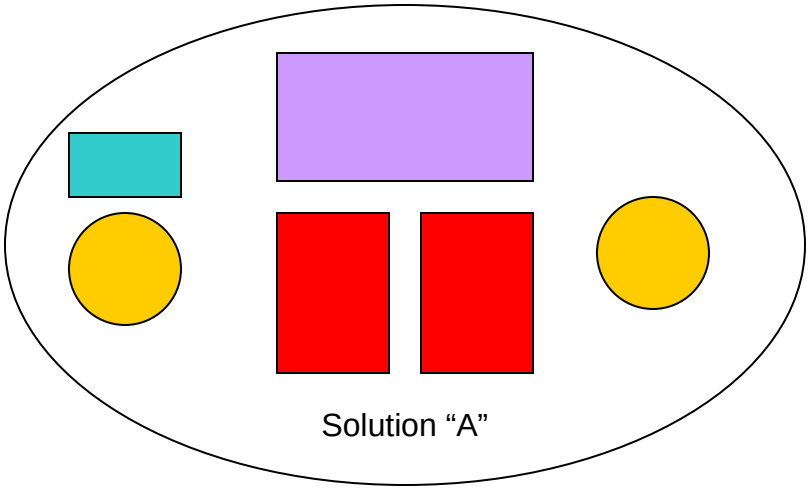
- Software costs come from making or buying software and from integrating, testing, maintaining and operating it. FPs are a superior way of modeling and comparing software costs – a uniform measure of normalized size

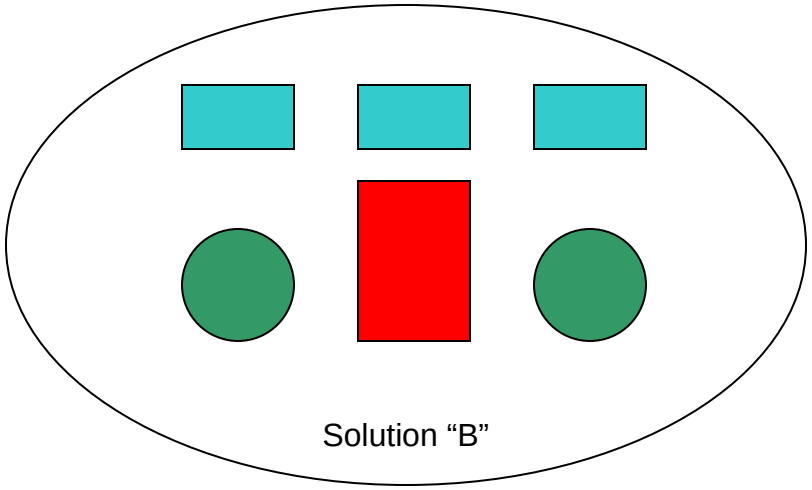
- To find the cost of something, we must establish its scope. FPA gives us a logical way of identifying a boundary around software. In SOA, not just entire applications but also individual software components, are developed in separate projects. Their size relates to effort, cost, schedule, etc.

- Measure FPs of components and associate the size with cost, effort, defects, etc. Measure at the level the work is being performed. This reflects the reality of development cost.

ROI in Component Based Software – Return (Value)

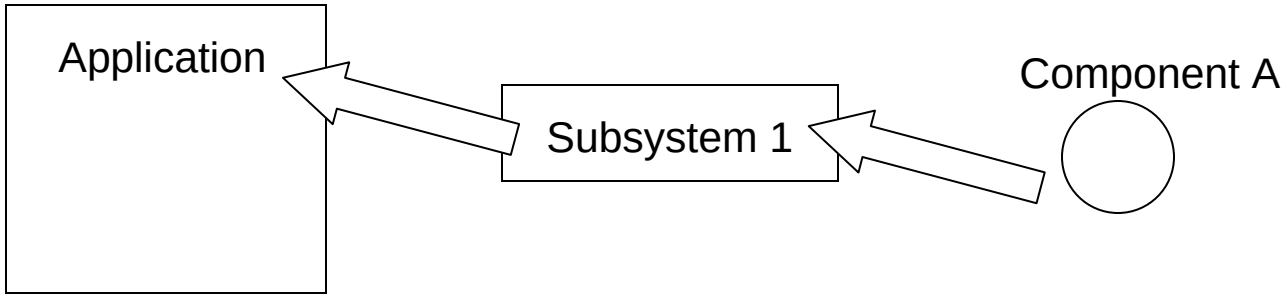
- Measure value at the level of an entire application or service
  - Components generally do not provide value by themselves
  - Value is obtained by the collective whole: application or service
- To provide a software solution and deliver value, different combinations of different components form different sources yield different collective costs. In this way, different results are produced in the ROI (or value/cost) expression



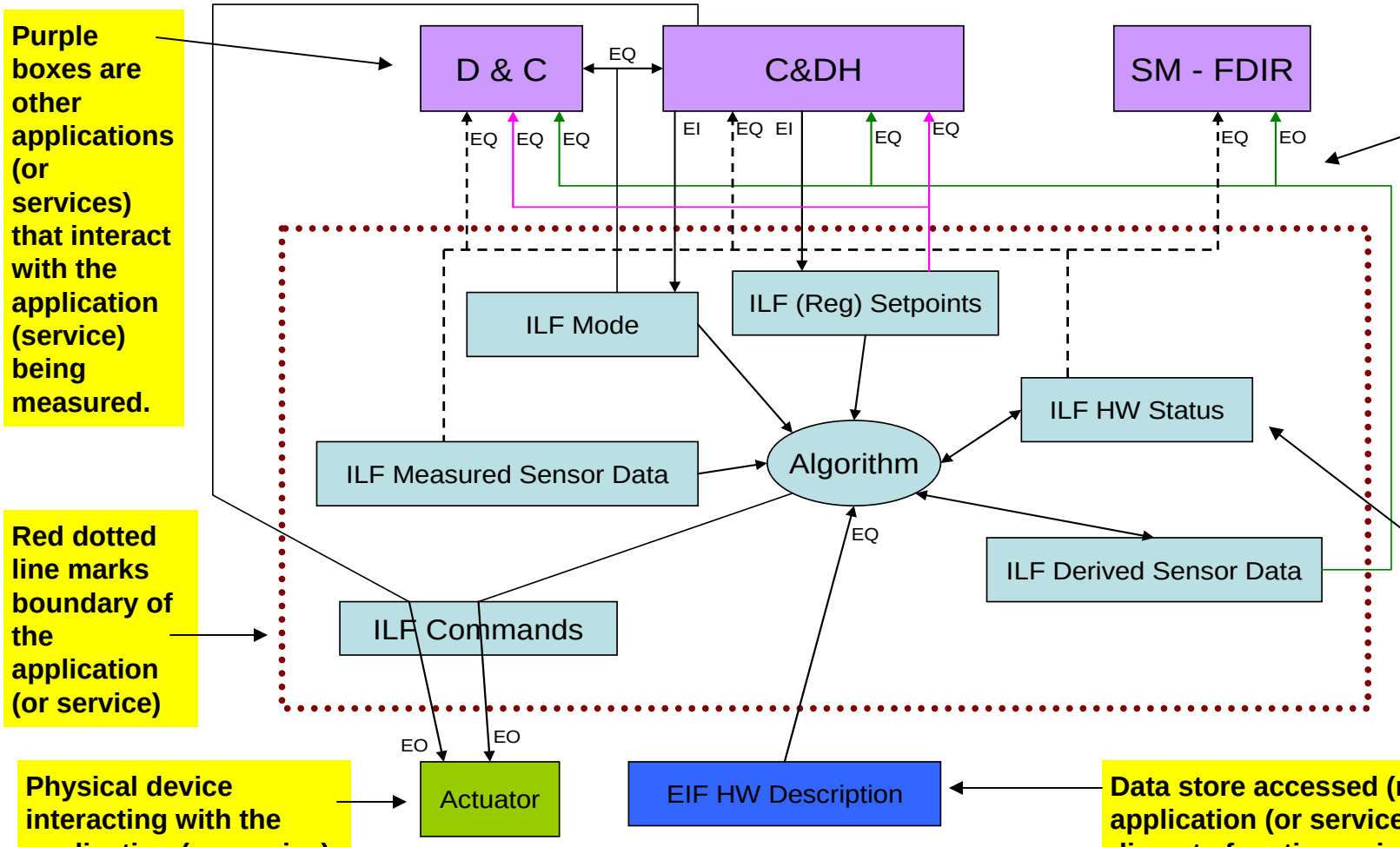


Approach to FPA on SOA

- The critical issue of interpretation: careful definition of the boundary of what is measured
  - Apply FPA to the problem of software build/buy analysis, at different levels according to what is to be delivered by the project team, vendor or other software source: entire application, major component, minor component
  - Development or integration of software is accomplished through a project where effort, time, cost and quality are factors
  - In SOA, services are produced through combinations of components, each having costs and values, alone and in combination, at different times and circumstances
  - FPA works in this context provided that the correct view is taken of the user
  - The “user” in SOA is frequently another piece of software at a higher level
  - User functionality exists at multiple levels depending on the context



FPA Diagram Example – Individual Component "Z"
Data & Transaction Diagram



Purple boxes are other applications (or services) that interact with the application (service) being measured.

Other lines mark transaction flows (messages) to and from the application (or service). These transactions are identified and are counted as discrete function point components.

Red dotted line marks boundary of the application (or service)

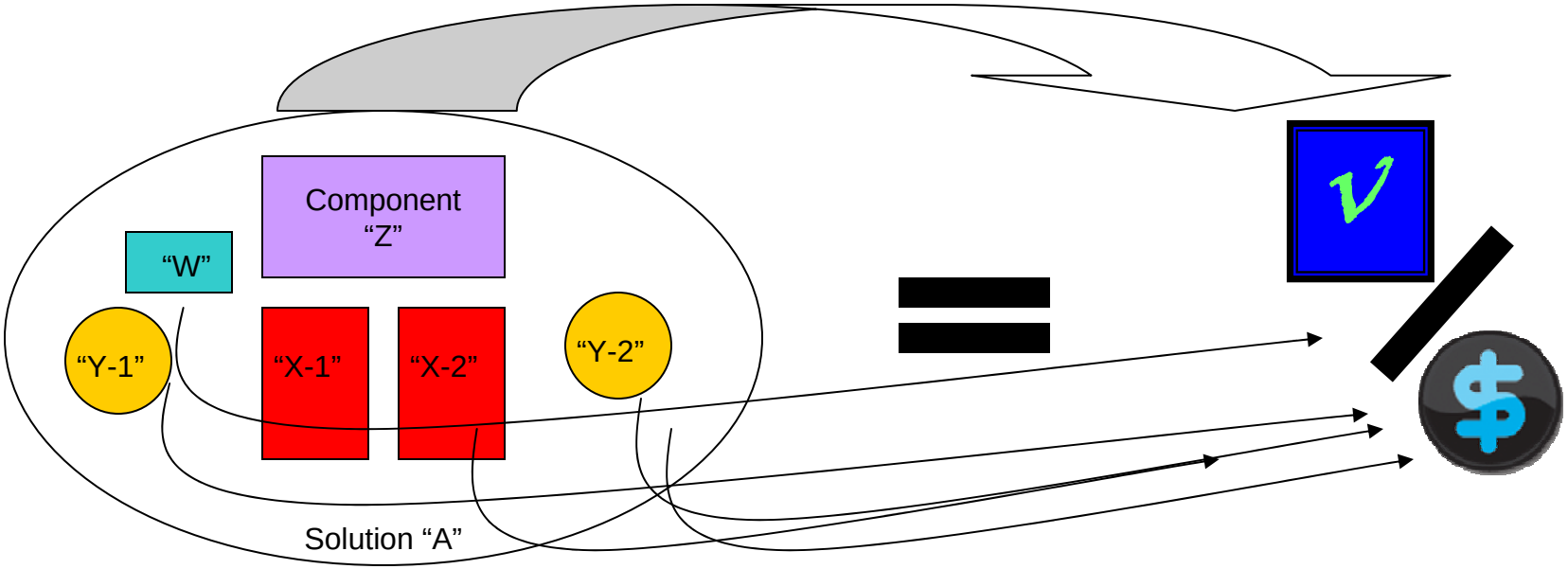
Blue boxes are data stores maintained by the application (or service) and are counted as discrete function point components.

Physical device interacting with the application (or service)

Data store accessed (read only) by the application (or service) and counted as a discrete function point component

Application / Service Composed of Component Software

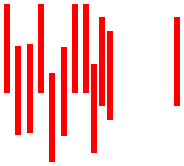
- Measure value at the level of an entire application or service
  - Value is obtained by the collective whole: application or service
- Measure the size of the entire application / service as a single object
- Accumulate the costs of each component individually



Major Factors in ROI for SOA

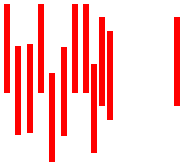
- ROI conditions
  - Increasing ROI is realized within the SOA environment after several projects have been implemented, each adding additional services to the pool of those available.
  - There must be sufficiently abundant opportunities to use shared services so that they can reduce the cost of producing applications.
  - There must be organizational standards to enable effective development and use of shared services. Ideally, industry standards will also arise.

- Costs
  - The upfront and ongoing costs for implementing SOA along with the first project utilizing the SOA is much greater than if the project is implemented within the current architectural concept
    - Each component built to be reusable is more expensive than an equivalent non-reusable component.
    - Each existing service incurs additional cost to make it reusable as a shared service
    - High quality requirements for new service components (more stringent than for non-reusable software) require more test and QA effort



Major Factors in ROI for SOA

- Benefits
  - SOA environment is expected to reduce the acquisition costs for each additional project – once the SOA environment has matured past the point that most new services must be made to be reusable.
  - Each additional project adds to the number of available shared services within the SOA pool
  - The number of shared services grows over time
  - Reduced development costs and schedules for projects using SOA architecture and components by acquiring pre-built capabilities
  - Lowered maintenance costs and consequently the total cost of ownership through fewer “instances” of function and fewer software licenses
  - Acquisition time required for each project is shortened due to the availability of shared services
  - Costs for each project are reduced due to the availability of shared services
  - The high quality characteristics of SOA software are propagated which reduces maintenance effort and cost



Major Factors in ROI for SOA

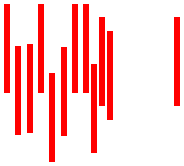
- Values and Motivations
  - Advantages of replacing custom crafted inflexible stove-piped systems with new systems built with common components
  - Shift from a “need-to-know” paradigm to a “need-to-share” paradigm
  - Standards for SOA will make it much easier to share and reuse components
- Obstacles
  - Lack of historical cost data proving advantages of component-based software solutions
  - Lack of component standardization and technical maturity
  - High visibility of SOA and related reusability approaches and urgency of need for solutions
  - Hype

SOA Software

- SOA implications for software – Positive Aspects
  - Continuously expanding collection of service capabilities – growth over time of reusable functional components
    - Less software to maintain
    - Reduced cost and reduced time to delivery
    - Maintenance - Reduction in the volume of software yields an equivalent effect on the amount of maintenance needed.
    - Idealized objective of SOA - reach a point where the cost of creating the next application is zero (or very close to zero).

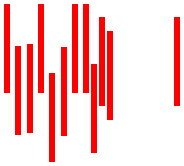
SOA Software

- SOA implications for software – Tradeoff Considerations
  - Compared to typical software components, the basic software building blocks of SOA (services) are usually much larger units of functionality than traditional functions or classes. Tradeoffs exist between efficient application production and efficient application execution in choosing the granularity of services:
    - The larger the components, the fewer the interface points required to implement any given set of functionality. Larger components with more functionality may be more efficient in terms of production and delivery of functionality.
    - Very large services may be bloated with unused functionalty since they are more likely to include functional capabilities that are not needed by all service users, making them relatively more difficult to maintain and adapt than smaller apps.
    - Very small services may be inefficient to use to develop applications because they require more work to orchestrate to build complete applications. They may also be inefficient in execution due to the greater total number of interface points required to implement any given set of functionality and consequently the greater total number of messages passed between small services in a complete app.
    - The smaller the services, the more precise the functional content, easing maintenance and adaptability.



SOA Software

- SOA implications for software – Negative Aspects
  - Different security levels and protocols complicate solutions and may affect suitability of some components for applications with restrictive security requirements.
  - SOA services must be usable without knowledge of, or access to, the internal processing of components. To accomplish this requirement some services that were not initially developed to be reusable must be modified.
  - Reusable software must have higher than typical quality level characteristics, requiring proportionately more testing and QA
  - Reusable software is more expensive than equivalent non-reusable software due to proportionately more testing and QA
  - Governance – need to control and manage changes to functionality with multiple uses/users/missions. This is needed to balance between changes needed to suit a few users versus code costs to be borne by all users. It is a question of values and tradeoffs. Governance requires effort and adds to cost and schedule
  - Discrepancies and incompatibilities in database contents and schemas will not be eliminated simply by adoption of SOA



ROI Issues for SOA

- Determination of Non-Financial Benefits
  - Benefits based on time to availability, quality (in many dimensions), and other factors such as customer satisfaction, business advantage, etc. will continue to be difficult to measure.
  - To identify and prioritize objectives of a software applications development or acquisition initiative, a technique known as Goal-Question-Measure (GQM) can be employed.
  - GQM enables a consensus approach to hierarchically define a system of measurement applicable to a specific organization or situation through top down decomposition.
  - The measurement system typically includes non-monetary as well as financial goals. Non-financial benefits can thus be identified and measured and incorporated into an ROI analysis that reaches beyond cost savings alone.
  - Benefits to users through expanded access to functionality and data, and benefits to implementing organizations through achievement of strategic and tactical goals such as adaptability, flexibility and resilience, can be valued through this method.
  - Non-financial benefits may be the decisive factors in any comprehensive analysis of ROI for SOA.

Contact Information

Terry Vogt
Certified Function Point Specialist
Booz Allen Hamilton
Associate
McLean, VA
(703) 377-4567
Vogt_terry@bah.com