Operationalizing the Architecture

Turning the Enterprise Architecture into an Active Mission Asset

John McDowall, Ph.D.
BAE Systems, Inc.
Enterprise Architecture has Failed

https://www.futurebanktoday.net/videoblog/episode-11-innovation-and-the-failure-of-enterprise-architecture


https://www.forbes.com/sites/jasonbloomberg/2014/07/11/is-enterprise-architecture-completely-broken/#3d448ad73710

But why?
Why EA Failed #1: Flawed Frameworks

• 1987: Zachman Architecture Framework (ZAF) was born as a system architecture framework
  • Enterprises have changed since 1987
  • Software development has changed even more
    • Services
    • Agile
    • DevOps
• All common EA frameworks descend from Zachman
  • All assume top-down design
  • All assume centralized control
Why EA Failed #2: Development Cycles

• POM cycle: 2 years
  • Every department and many agencies have independent budgets
  • No control of $$ = no conformance

• Fielding cycle: 3-7+ years (AFTER development!)
  • 10 years or more before a change in EA is fully implemented in the field
  • This dooms traditional EA approaches

Bonus: No effective conformance strategy
Consequences of Failure

• Interoperability problems persist

• Little data / interface standardization
  • Limited visibility into available interfaces
  • Mandated specifications rarely implemented

• No understanding of enterprise data flow / use

The patient is not getting better
A Recipe for Success

• Transform architecture from a design activity to an operational management activity
  • Monitor behavior
  • Test for conformance
  • Adjust to changing conditions

• Stop trying to understand implementation details
  • Establish measurable & enforceable enterprise goals
  • Publish rules & constraints
  • Monitor and adjust

The enterprise is a Complex Adaptive System—TREAT IT THAT WAY!
Characteristics of Complex Adaptive Systems

• The whole is more than the sum of the parts
  • Each of the parts may be perfectly understood
  • Behavior of the whole is not readily predictable

• Emergent behaviors arise from many independent actors

• Respond to changes in the environment
  • Changes in individual behavior
  • Changes in external factors
Example: Ant Hill

- No ant can do math
- An ant colony can find the shortest path between the ant hill and every food source
- Humans could not do this efficiently until Edsger Dijkstra published his algorithm in 1959
The DoD Enterprise is a CAS

- Department of Defense
  - 4 Services
  - 25+ Agencies / Field Activities
  - 4M+ individuals
  - 1000s of systems

- Examples of emergent behaviors in the enterprise
  - Users employing a system differently than expected
  - Workarounds to bypass a problematic system

The enterprise will evolve faster than we can design it
Proof of Concept

• Created on JWICS
  • Using existing data
    • NO MANUAL DATA CALLS!
    • No additional data gathering
    • Using currently deployed monitoring tools (RMF)
  • Using existing software
    • COTS
    • Open Source

• Implemented on unclass for demonstration purposes (notional data based on operational data)
Insight Into Operational System Behavior

Notional data based on actual systems
Visibility Into Enterprise Data Flow

Notional data based on actual systems
Details About Data Producers and Consumers

Notional data based on actual systems
Summary

• We need to reform the practice of enterprise architecture
  • Stop focusing on paperwork
  • Focus on operations instead of designs

• Enforce through continuous monitoring & testing
  • Observe and measure mission capabilities / user behavior
  • Establish testable conformance criteria
  • Apply resources to solving observed problems

We can make EA an effective operational tool
Questions

john.mcdowall3@baesystems.com