# CROSS-PROGRAM WEAPONS SYSTEM SOFTWARE ACQUISITION CAN SAVE BILLIONS

Rick Brennan Operational Systems, Inc May 17, 2012

### SOFTWARE IN MILITARY AIRCRAFT



Source: NASA Study on Flight Software Complexity

#### Physical Goods Cost Behavior

- Notional example:
  - \$40 million for development, integration, test, certification, and other non-recurring costs
  - \$10 million for manufacturing, test, distribution, integration, and other recurring, per-unit costs
  - 1 core capability delivered
  - 50 units built



#### Software Cost Behavior Comparison

Notional Example:

- \$40 million for development, integration, test, certification, and other non-recurring costs
- \$0 for manufacturing, test, distribution, integration, and other recurring, per-unit costs
- 1 core capability delivered
- 50 initial software instances built



# Economic Effect of Stovepiping

Notional Example:

- Each program builds a unique hardware and software product
- \$40 million fixed development cost for each hardware product
- \$10 million marginal manufacturing cost per hardware unit
- \$40 million fixed development cost for each software product
- \$5 million fixed development cost for each software version after V1.0
- \$0 marginal manufacturing cost per software instance



#### Stovepipe Economic Effects Compounded by Software Improvement Cycles Stovepiper

Notional Example:

- Each program builds a unique software product
- \$40 million fixed development cost for each software product
- \$5 million fixed development cost for each software version after V1.0
- \$0 marginal manufacturing cost per software instance

Shared Intellectual Property



# THE POWER OF SHARED INTELLECTUAL PROPERTY

- In the best case, once fixed costs for development, test, certification, and verification are paid, the cost to scale software is close to zero
- Someone has to pay those up front fixed costs

But Everyone Doesn't!

# CURRENT BARRIERS TO SHARING IP ACROSS PROGRAM BOUNDARIES

- Requirements development and management structure
- Funding structure
- Cultural interpretation of risk
- Business model
- Shortage of Government Lead Systems Integrator basic knowledge and skills
- Systems Command structure
- Systems Command skillset gaps

# MINIMIZING SHARING COSTS

- Identify appropriate intellectual property sharing systems targets
- Facilitate cross-program coordinated software development
  - Cooperative Product Line applications development
  - Community requirements management for frameworks and tools
- Shared Resources
  - Shared open source software infrastructure
  - Shared, cloud-based development and test environment
- Policy optimization
  - Financial incentives for programs that cooperate and collaborate
  - Re-thinking of the concept of risk
  - Streamlined IA and safety certification
- Business Model update
  - Separate physical vehicle acquisition from software applications acquisition

# QUESTIONS?