CROSS-PROGRAM WEAPONS SYSTEM SOFTWARE ACQUISITION CAN SAVE BILLIONS

Rick Brennan
Operational Systems, Inc
May 17, 2012
### Cross-Program Weapons System Software Acquisition Can Save Billions

**1. REPORT DATE**
17 MAY 2012

**2. REPORT TYPE**

**3. DATES COVERED**
00-00-2012 to 00-00-2012

**4. TITLE AND SUBTITLE**
Cross-Program Weapons System Software Acquisition Can Save Billions

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**6. AUTHOR(S)**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Operational Systems, Inc, 1231 Collier Road NW, Suite D, Atlanta, GA, 30318-2313

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

**10. SPONSOR/MONITOR’S ACRONYM(S)**

**11. SPONSOR/MONITOR’S REPORT NUMBER(S)**

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution unlimited

**13. SUPPLEMENTARY NOTES**
Presented at the 9th Annual Acquisition Research Symposium, May 16 - 17, 2012, Monterey, CA. U.S. Government or Federal Rights License

**14. ABSTRACT**

**15. SUBJECT TERMS**

**16. SECURITY CLASSIFICATION OF:**

<table>
<thead>
<tr>
<th>a. REPORT</th>
<th>b. ABSTRACT</th>
<th>c. THIS PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>unclassified</td>
<td>unclassified</td>
<td>unclassified</td>
</tr>
</tbody>
</table>

**17. LIMITATION OF ABSTRACT**
Same as Report (SAR)

**18. NUMBER OF PAGES**
10

**19a. NAME OF RESPONSIBLE PERSON**

---

Standard Form 298 (Rev. 8-98)

Proscribed by ANSI Std Z39-18
SOFTWARE IN MILITARY AIRCRAFT

Functionality provided to the aircrew by software

Year of introduction

1960 (F-4)  1964 (A-7)  1970 (F-111)  1975 (F-15)  1982 (F-16)  1990 (B-2)  2000 (F-22)

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

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

Cost of Stovepiped Software Development

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?