System Integration with Commercial Off-the-Shelf Software (COTS Integration): Overview of Issues

Barry Boehm, USC-CSE
CSE Executive Workshop
March 12, 1997

Outline

• CSE Affiliates’ Appreciation
• What Is (and Isn’t) COTS?
• COTS Payoffs: Hardware-Software Trends
• COTS Phenomena, Pitfalls, and Principles
• COTS Integration: Executive-Level Issues
USC-CSE Affiliates (28)

- Commercial Industry (10)
  - Bellcore, EDS, IDE, Lucent, Motorola, Network Programs, Rational, Sun, TI Xerox
- Aerospace Industry (10)
  - Allied Signal, Boeing/Rockwell, GDE Systems, Hughes, Litton, Lockheed Martin/Loral, Northrop Grumman, Raytheon/E-Systems, SAIC, TRW
- Government (3)
  - AFCAA, USAF Rome Lab, US Army Research Labs
- FFRDC’s and Consortia (5)
  - Aerospace, IDA, MCC, SEL, SPC

Affiliates Via Mergers

- Sanders, RCA Aero, GE Aero, GD-FW, Loral -- Lockheed Martin
- Ford Aero, LTV, IBM-FSD, Unisys DSD -- Loral
- Westinghouse Aero -- Northrop Grumman
- E-Systems, Hughes, TI -- Raytheon
- Teledyne Aero, PRC -- Litton
- Rockwell, MDAC -- Boeing
- Bellcore -- SAIC
What is (and Isn’t) COTS?

- terms from recent Ground System Architectures Workshop

- COTS: Commercial Off-the-Shelf
- GOTS: Government Off-the-Shelf
- HOTS: Hot Off-the-Shelf
- NOTS: Not Off-the-Shelf
- ROTS: Research Off-the-Shelf

Outline

- CSE Affiliates’ Appreciation
- What Is (and Isn’t) COTS?
- COTS Payoffs: Hardware-Software Trends
- COTS Phenomena, Pitfalls, and Principles
- COTS Integration: Executive-Level Issues
Hardware-Software Trends

Issues

• Why does software look bad?
  – Curves count different things
• Suppose we counted similar things?
  – Transistors, etc. in service
  – Lines of code in service
• This makes software look comparable to hardware
  – Largely due to COTS

COTS Phenomena, Pitfalls and Practices

• You have no control over a COTS product’s functionality or performance.
• Most COTS products are not designed to interoperate with each other.
• You have no control over a COTS product’s evolution.
• COTS vendor behavior varies widely.
1. You have no control over a COTS product's functionality or performance.

**Pitfalls to Avoid**
- Using the waterfall model on a COTS integration project.
- Using evolutionary development with the assumption that every undesired feature can be changed to fit your needs.
- Believing that advertised COTS capabilities are real.

**Practices to Adopt**
- Use risk management and risk-driven spiral-type process models.
- Perform the equivalent of a "receiving inspection" upon initial COTS receipt.
- Keep requirements negotiable until the system's architecture and COTS choices stabilize.
- Involve all key stakeholders in critical COTS decisions.

2. Most COTS products are not designed to interoperate with each other.

**Pitfalls to Avoid**
- Premature commitment to incompatible combinations of COTS products.
- Trying to integrate too many incompatible COTS products.
- Deferring COTS integration till the end of the development cycle.
- Committing to a tightly-coupled subset of COTS products with closed, proprietary interfaces.

**Practices to Adopt**
- Use the Life Cycle Architecture milestone as a process anchor point.
- Use the Architecture Review Board (ARB) best commercial practice at the Life Cycle Architecture milestone.
- Go for open architectures and COTS substitutability.
3. You have no control over a COTS product’s evolution.

Pitfalls to Avoid
- “Snapshot” requirements specs and corresponding point-solution architectures.
- Understaffing for software maintenance.
- Tightly coupled, independently evolving COTS products.
- Assuming that uncontrollable COTS evolution is just a maintenance problem.

Practices to Adopt
- Stick with dominant commercial standards.
- Use likely future system and product line needs as well as current needs as COTS selection criteria.
- Use flexible architectures facilitating adaptation to change.
- Carefully evaluate COTS vendors’ track records with respect to predictability of product evolution.
- Establish a pro-active system release strategy, synchronizing COTS upgrades with system release.

4. COTS vendor behavior varies widely

Pitfalls to Avoid
- Uncritically accepting COTS vendors’ statements about product capabilities and support.
- Lack of fallbacks or contingency plans.
- Assuming that an initial vendor support honeymoon will last forever.

Practices to Adopt
- Perform extensive evaluation and reference-checking of a COTS vendor’s advertised capabilities and support track record.
- Establish strategic partnerships or other incentives for COTS vendors to provide support.
- Negotiate and document critical vendor support agreements.
Outline

- CSE Affiliates’ Appreciation
- What Is (and Isn’t) COTS?
- COTS Payoffs: Hardware-Software Trends
- COTS Phenomena, Pitfalls, and Principles
- COTS Integration: Executive-Level Issues

Some Executive-Level Issues

- Integration of COTS with corporate strategies
  - E.g., 10X cycle time, six-sigma quality
- Effects of COTS on corporate software processes
  - Requirements-to-capabilities or vice-versa?
- Corporate strategies vis-a-vis COTS vendor
  - Strategic partnerships? Substitutability?
- Assessing emerging technology and standards?
  - CORBA vs ActiveX? RDB vs OODB? OO everything?
Some Additional Questions:
Larry Bernstein

- Balance Sheet: Expenses vs. cost-of-good?
- Effects on bottom-line profitability?
- Adapting to discontinued versions or products?
- "Sharing" intellectual property?
- Stimulating COTS commonality across the organization?
- Corporate product line evolution and COTS evolution?
- Most important criteria for evaluating COTS?

Architecture in a Project's Life Cycle
It encompasses the requirements, architecture and high level design phases of the typical waterfall diagram. It also continues throughout the life of the project (someone continues to wear the architect's hat).