Managing Commercial-Off-the-Shelf (COTS) Integration for High Integrity Systems: How Far Have We Come?
Problems and Solutions in 2003

Karen Owens, Suellen Eslinger, Geri Chaudhri
The Aerospace Corporation
Breakout Session Goals

- View the topic of COTS integration for high integrity systems from multiple perspectives
  - Cost
  - Management
  - Architecture
  - Integrator
  - Vendor
  - User
- Identify solutions and successes
  - What factors contributed to the successes?
- Identify ongoing and new problems and issues
  - What factors contributed to the problems?
  - How can the problems be reduced or eliminated in the future?
- Update survey on COTS upgrade release frequency
What Happened

- **Participant introductions**
  - 31 Session participants included Aerospace, industry, academia, acquirers, users, and cost estimators

- **Management perspectives**
  - “Quantitative Management of COTS-Based Systems: The Role of Cost Estimation” - Marilee Wheaton, Aerospace, Steven Wong, Northrop-Grumman (heritage TRW)
  - “Managing COTS Integration for High Integrity Systems: Observations from the COCOTS Database” – Betsy Clark, Software Metrics, Inc.

- **Technical perspectives**
  - “The Role of Architecture in Managing COTS-Based High Integrity Systems” – Rodney Davis, Command and Control Technology
  - “COTS or Development: Simulation Tools for Ground Systems Integration” – Tom Tillman, L-3 Communications

- **Lively discussion**
- **Collected data for COTS upgrade release survey**
The Role of Cost Estimation –
Marilee Wheaton and Steven Wong

• Use CMMI Practices for Estimating Cost and Schedule
  ❖ Organizational Process Performance, Quantitative Project Management

• Build and Use COTS Model and Data Baselines
  ❖ Number of COTS packages, COTS breakage, volatility, interdependency, defect reports

• Meet COTS Estimating Challenges
  ❖ Scoping functionality, productivity, limited data
  ❖ Separating COTS effort from development

• Collect data on COTS Integration Activities
  ❖ Assessment, Understanding, Tailoring, Glue (COCOTS and SEER-SEM)

• Keep estimates current: best, worst and expected
  ❖ Use estimate data for management (e.g., earned value, EAC)
Managing COTS for High Integrity Systems – COCOTS Database Observations - Betsy Clark

- “COTS products are associated with some risk” – Vic Basili
- Types of products for 11 safety critical, real-time systems
  - Operating systems, GUI generator, DBMS, network management, communications protocols, disk array
- Attributes considered in evaluating COTS
  - Performance, interoperability, robustness
- Reliability Solutions:
  - Fault-tolerant architectures
  - Detailed evaluations
  - Mature components
  - Purchase of source code
  - Agreement requiring 24-hour responses to critical problems
- Maintenance Solutions:
  - Focus on critical components
  - Wrappers
  - Freezing configuration
COTS or Development: Simulation Tools – Tom Tillman

• Need satellite and range data for Command and Control Ground System development and integration that
  - Is realistic, easily accessible, and affordable
  - Contains anomaly conditions
  - Supports mission scenario flexibility and frequent changes

• Solutions:
  - Simulator that allows varying satellite date
    - Measurand ranges, anomalies, reusable databases for satellite families
  - Flexible COTS simulation controls
    - Commands from ground system
    - Time control, checkpoint and restarts
    - Configurations via databases instead of software
  - Ongoing training through simulation
  - Cost Effective
The Role of Architecture - Rod Davis

- **Problems:**
  - Loss of Control
  - Discontinuities in understanding whole system
  - Complexity with many components

- **Architectural Solutions:**
  - Build understanding through evaluation and qualification; apply influence in the market
  - Postpone detailed decisions until architecture foundations are set
  - Use open standards
  - Minimize interconnections
  - Architect and engineer for security and reliability – use patterns
  - Analysis of Alternatives
  - Document architecture implications of decisions to support evolving architecture

“Apply good design practices to COTS integration”
Conclusion

• CBS development, integration, and sustainment have inherent uncertainties beyond the control of the acquirers, developers and users

• Effective CBS development and sustainment requires a change of processes and attitudes across the entire life cycle and among all parties

  - Acquirers
  - Costers
  - Architects
  - Developers
  - Maintainers

  - Users
  - Integrators
  - Procurement
  - Contracts
Survey on COTS Upgrade Release Frequency

1. In your experience, what is the average duration between releases of a given COTS Product?
   - GSAW 99 = 6.3 months
   - GSAW 00 = 8.5 months
   - GSAW 01 = 8.75 months
   - GSAW 02 = 9.6 months (range: 6 to 18)
   - GSAW 03 = 11.2 months (range: 2 to 24) (9 data points)

2. For system(s) with which you have experience, how frequently are system upgrades released that incorporate COTS software upgrades?
   - GSAW 02 = 20.5 months (range: 4 to 70)
   - GSAW 03 = 16.5 months (range: 6 to 36) (9 data points)
Survey on number of COTS products

3. For system(s) with which you have experience, how many COTS products are in your system?
   - GSAW 03 = 37 (range: 1 to 150) (9 data points)

4. For system(s) with which you have experience, what percentage of COTS products are upgraded?
   - GSAW 03 = 58% (range: 5% to 100%) (9 data points)
Key for following charts:
- “Gn” indicates GSAW 2001 issue and ranking
- “A” indicates 2000 Aerospace study issue
New Issues

- Is COTS usage really cost effective?
- When and if to upgrade
- How to capture lessons learned – Can product reviews be shared?
- How is cost of maintenance measured?
- Integrating COTS into project lifecycle particularly spiral development process
- Technology exportability issues
  - Maintenance, requirements traceability
- COTS products within the development environment
Highest Scoring “Top 3” Issues (13 ballots)

- Accurately costing all aspects of CBS development and maintenance
- Incompatibilities among COTS products
- Processes for trading cost, schedule, requirements, and O&M concepts against COTS capabilities
- Requirements vs. COTS capabilities
- Adverse effects of product upgrades on system
- Rapid technology turnover and limited support of past releases
- Integration of multiple COTS products
- Cost vs. benefit of upgrading
- Dropped of de-emphasized platforms and products
“COTS Survey” – Participants 2003 and Previous Years

• Be sure to include all COTS-related costs, especially those not included in the cost model estimates (e.g., licenses, training)
• Expect glue code to have a lower productivity than custom software
• Cost of COTS versus custom development needs to be evaluated for the full life cycle (not just development)
  ❖ Distribution of costs will be different for COTS and custom development projects

“Despite discussion to the contrary, using COTS is still cheaper and more effective than building”
Emerging Issues from Kohl

• COTS ‘certification’
  - What is it? Approaches to achieve it?
  - How to measure or validate it?
  - As compared to custom built software ‘certification’

• COTS content at major milestone reviews
  - SRR, SDR, PDR, CDR, TRR, etc
  - What content should be presented at each review?

• COTS impacts to lifecycle processes
  - Changes to existing processes (requirements, evaluation)
  - Differences in sequence of activities
  - Milestone review impacts (see 2nd bullet)
Product Issues

• Requirements vs. COTS capabilities (G1)
• Integration of
  ❖ Multiple COTS products (G2)
    ➥ Incompatibilities among COTS products (A)
  ❖ COTS products with new/reuse software (G5)
• COTS independent architecture (G3)
  ❖ Designing architectures for COTS evolution (A)
  ❖ Designing in safety, security, supportability (A)
• Mission risk (G6)
• Cross platform portability (G11)
• Standards: good, bad, ugly? (G14)
• API breakage (“unplug and replay”) (G15)
• Dormant functionality or features (G18)
• Adverse effects of product upgrades on system (A)
Process Issues - Developer

• Robust initial and periodic COTS product evaluation (A)
• Prototyping in a system context (A)
• Testing in operational context (G12)
  ❖ Regression testing of upgrades in system context (A)
• Adapting software and systems engineering processes for CBS development and maintenance (A)
• Still need systems and software engineering (A)
• Need enhanced CM processes (A)
• Planning for COTS upgrades and evolution during development and maintenance (A)
• Selection of hardware platforms with availability of COTS software as key criterion (A)
Process Issues - Customer and User

• Acquisition and support strategies (G16)
• Adapting customer/user processes to CBS acquisition, operations and maintenance (A)
• Processes for trading cost, schedule, requirements, and O&M concepts (A) against COTS capabilities
  - Need requirements prioritization (A)
• Need contracts compatible with CBS development and maintenance (A)
• Standardized processes for safety certification and security accreditation of CBS needed (A)
• Standardized license processes to ensure suitability of licenses and maintaining currency (A)
Resource Issues

- Cost vs. benefit of upgrading (G7)
- Acquiring and maintaining CBS skills (G9)
- Accurately costing all aspects of CBS development and maintenance (A)
- Optimal scheduling of upgrades (A)
- Increased computer resources for upgrades (A)
- Modifying COTS is a BAD idea! (A)
- Need cost and schedule management reserves (A)
- Reallocating time and effort across life cycle (A)
  - More time for evaluation, prototyping and analysis (A)
  - Less time for implementation; more time for integration (A)
Marketplace Issues

- **Product maturity (G4)**
  - Dropped or de-emphasized platforms and products (A)
  - Changes in fees and fee structure for licenses and services (A)

- **Marketplace maturity (G8)**
  - Vendor volatility (A)

- **Vendor responsiveness (G17)**
  - Changes in type and quality of vendor support (A)

- **Definition of COTS (= “for sale”) (G19)**

- **Suitability of licenses for user application (e.g., expiring keys, export restrictions) (A)**

- **Release schedule, content and quality unpredictable (A)**

- **Rapid technology turnover and limited support of past releases (A)**
Intergroup Interaction Issues

• Customer resistance to COTS--NIH (G10)
• Excessive customer bias toward COTS (A)
• Vendor relationships (G13)
• Establishing and maintaining active partnership between customer, developer and user (A)
• Need flexible and efficient responses to unexpected impacts by customer/user (A)
Resources

• CeBASE COTS
  ❖ http://www.cebase.org/

• CeBASE COTS Lessons Learned site
  ❖ http://fc-md.umd.edu/ll/index.asp

• SEI’s COTS-Based Systems Initiative
  ❖ http://www.sei.cmu.edu/cbs/

• International Conference on COTS-Based Software Systems (ICCBSS - “ice cubes”)
  ❖ http://seg.iit.nrc.ca/iccbss/
Aerospace Publications


  ❖ Includes paper and briefing charts

