

# COTS Lessons Learned

---

Betsy Clark  
Software Metrics, Inc.

October 24, 2000

Sponsored by the Federal Aviation Administration's  
Software Engineering Resource Center

# COTS Lessons Learned

---

- At the conclusion of the data collection interviews, we asked about lessons learned
- Sixteen of the twenty projects responded
  - Twelve were FAA
    - Air Traffic Management: 6
    - Air-Ground Communications: 2
    - Support systems (non-operational): 2
    - Administrative: 2
  - Four were DoD
    - Missile Launch Control: 2
    - Mission Planning: 1
    - Administrative: 1
  - Large range in the number of COTS products integrated
    - 1 to 150

# Areas Addressed

---

- Problems with vendors (6)
- Need for flexibility in defining requirements (5)
- Importance of operational demonstrations (5)
- Assessment of specific product attributes (5)
- Life-cycle issues (5)
- COTS integrator experience (3)
- Training on COTS packages (2)
- Need for technology watch to keep up with vendors (2)
- Interface to legacy systems (1)
- Vendor management (1)
- Impacts of volatility during development (1)

# Problems with vendors (6)

---

*“Our biggest problem was with the vendors, especially with believing their claims. Very few components worked as advertised.”*

- Vendors promise and don't deliver
- Products don't work as advertised
  - one contractor is currently in litigation with an OS vendor
- Don't assume a quantity discount
  - Negotiate entire price up front

# Need for Flexibility in Defining Requirements (5)

---

- *“Distinguish between essential and negotiable requirements. Be flexible where you can.”*
- *“We spent 14 out of a total of 22 months iterating between requirements, business processes and the marketplace. This is something we did right.”*
- *“If you can bend your requirements, COTS is cheaper. Otherwise you’re better off with custom developed.”*

# Importance of Operational Demos (5)

---

*“Operational demos are important. At that point, vendors are bending over backwards to sell their components so they’ll participate.”*

*“Up-front time is critical. That’s when you have leverage with vendors. Once you buy their product, they are a lot less willing to help out.”*

*“We should have done operational demonstrations.”*

# Assessment of Specific Attributes (5)

---

*“If you have a safety-critical system, you don’t want state of the art COTS; you want mature components.”*

*“Never use an untried OS.”*

A number of projects expressed regret that they did not spend more time assessing:

- portability (2)
- inter-component compatibility (1)
- flexibility (of user interface) (1)
- installation ease (1)

# Life Cycle Issues (5)

---

- Maintenance Strategies
- Maintenance Costs

# Maintenance Strategies

---

- Supportability of COTS viewed as a major issue for safety-critical systems
  - Out of service is a critical problem
    - contractor purchased source code and will maintain COTS software
  - Operational disruption and expense to upgrade
    - How do you upgrade systems once they are in place and operating?
    - What is an effective strategy for upgrading?
      - Freeze and redo the system in 10 years?
      - Incorporate all versions from all vendors whenever they come out?
      - Refresh every 2 years?
      - Refresh a selected set of components every 2 years?

# Maintenance Strategies (cont)

---

- Should have an environment set up so you can load new versions onto the existing configuration and decide whether or not to upgrade.

# Maintenance Costs

---

- Several projects expressed the view that COTS saved money during development but shifted costs to operational side of the life cycle

*“People have to look at the entire life cycle realistically – not just the development cost but consider what it’s going to cost to maintain over a number of years.”*

*“We need to know how to estimate costs for the entire lifecycle.”*

# Integrator Experience (3)

---

*“Look carefully at the credentials of the COTS integrators. There is a tendency to oversell.”*

- Lack of experience
  - in COTS integration
  - with specific products
  - in government domain

# Need for technology watch (2)

---

*“You have to constantly monitor the state of the COTS components. We had a company fold and we were taken by surprise. A technology watch would have prevented us from getting stuck.”*

# Need for training on COTS packages (2)

---

- *“We should have planned 6 months of training for those doing the tailoring.”*
- *“We spent a month just becoming familiar enough with the COTS packages to use them.”*

# Interface to Legacy Systems (1)

---

*“The achilles heel of all COTS projects is the interface to legacy systems. They fail here over and over again. This is the part that is not working well for us.”*

- data differences
- transaction differences

# Vendor Management (1)

---

*“We spent 65 staff months with the vendor. We had never planned on this.”*

# Impacts of Volatility During Development (1)

---

- *“You may have to re-tailor COTS components with new releases. In our case, we had to write new scripts to accommodate new features.”*

# General Comments

---

- With COTS, you are buying into a different process
  - operational demos are critical
  - requirements must be flexible
  - costs shift to the right
- *COTS is high risk because we are dependent on someone else. There needs to be a process to help people evaluate their risks.*