GSAW 2004

Best Practices for the Acquisition of COTS-Based Software Systems (CBSS): Experiences from the Space Systems Domain

Richard J. Adams and Suellen Eslinger
Software Acquisition and Process Office
Software Engineering Subdivision
The Aerospace Corporation

THE AEROSPACE CORPORATION

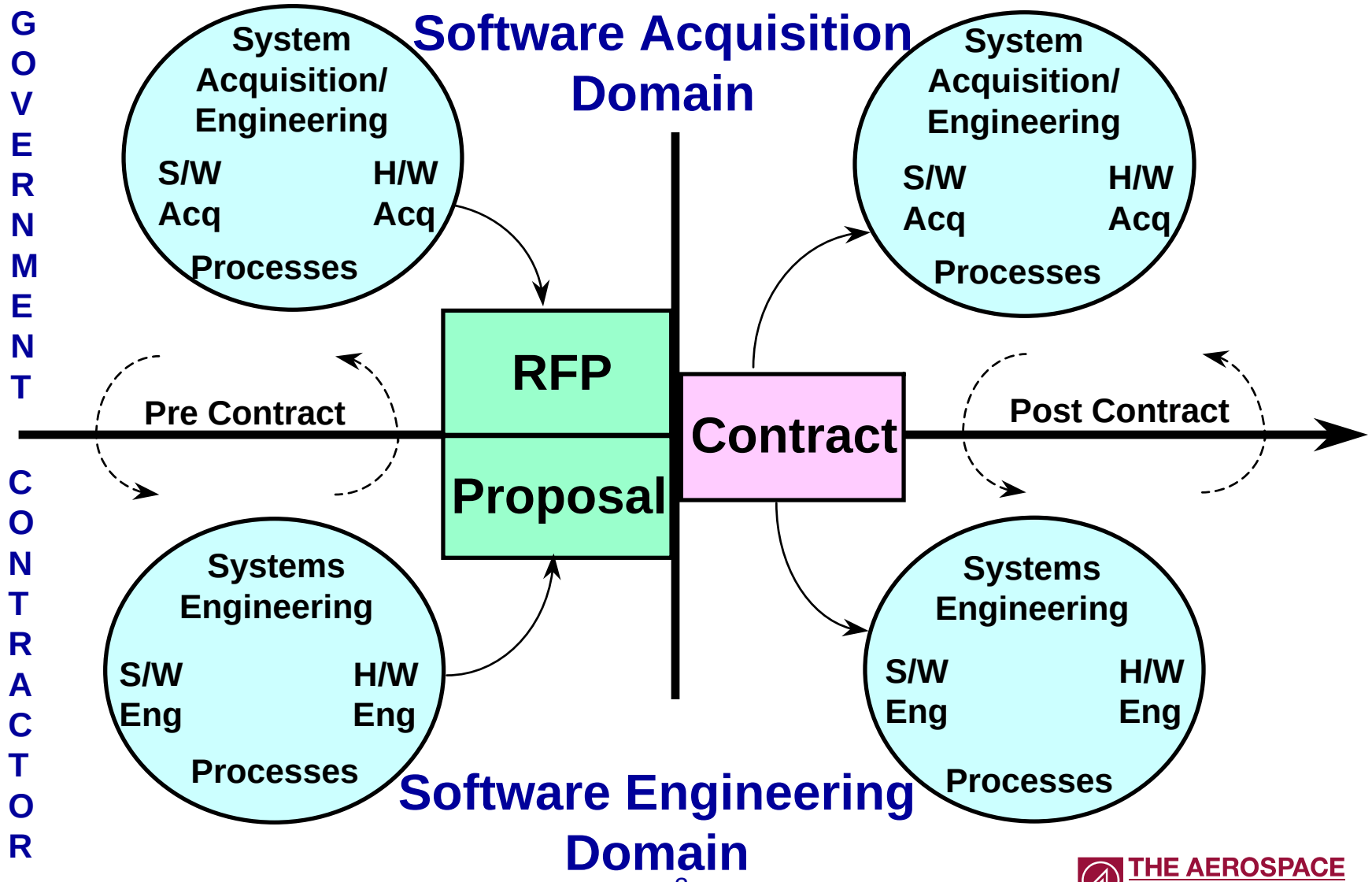
Acknowledgements

This work would not have been possible without assistance from the following:

- **Reviewers and Sponsor**
  - Linda A. Abelson, Software Acquisition and Process Office
  - Peter Hantos, Software Acquisition and Process Office
  - Karen L. Owens, Software Acquisition and Process Office
  - Mary A. Rich, Principal Director, Software Engineering Subdivision
  - Michael Zambrana, USAF Space and Missile Systems Center, Directorate of Systems Engineering (Sponsor)

- **Funding source**
  - Mission-Oriented Investigation and Experimentation (MOIE) Research Program (Software Acquisition Task)

Software Acquisition vs. Software Engineering



Characteristics of DoD Space Systems

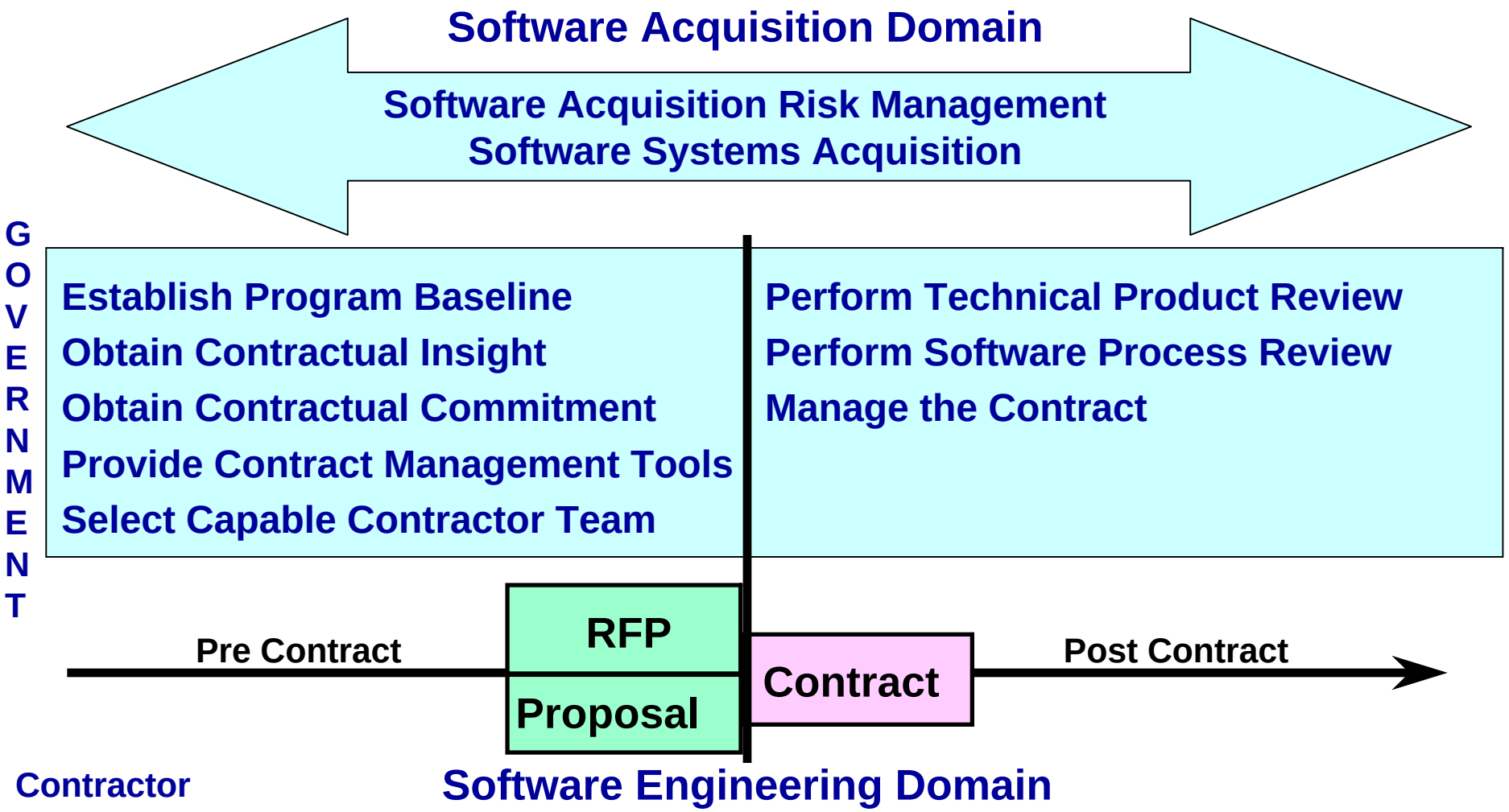
- **Large software-intensive systems**
  - SLOC order of magnitude: 10^5 onboard and 10^6 – 10^7 on the ground
  - Multi-satellite constellations
  - Multiple ground elements, frequently worldwide
- **Complex combinations of hardware and software**
- **Complex external and internal interfaces**
- **Usually unprecedented**
- **High reliability and integrity requirements**
- **Increasingly large number of COTS software products integrated into the ground systems**

Space Domain CBSS Acquisition Best Practices must support these characteristics.

Space Domain CBSS Lessons Learned

1. Critical aspects of CBSS development and sustainment are out of the control of the customer, developer and user.
2. Full application of system and software engineering is required throughout the CBSS life cycle.
3. CBSS development & sustainment require a close, active & continuous partnership among the customer, developer & user.
4. Every CBSS requires continuous evolution throughout development and sustainment.
5. Current processes must be adapted for CBSS acquisition, development and sustainment.
6. Actual cost and schedule savings with CBSS development and sustainment are overstated.

Software Acquisition Best Practice Roadmap



Software Acquisition Best Practices for Establishing the Program Baseline

Perform software architecture trade studies

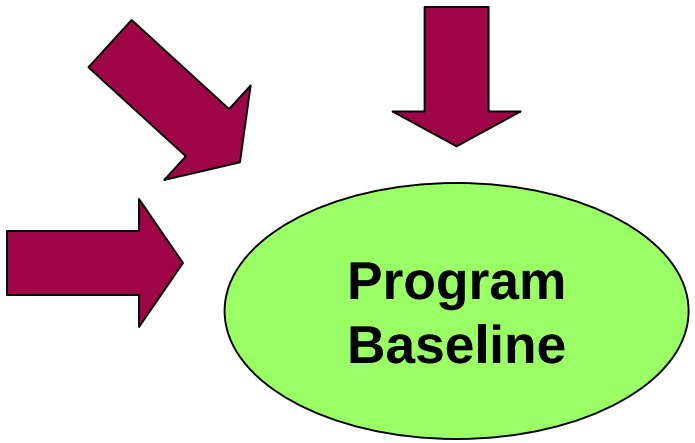
- With system architecture trades
- Include major COTS & legacy components
- Supports Government software architecture baseline selection
- Include user in all trades

Include software in system performance requirements

- Prioritized requirements
- COTS software support requirements
- Specialty engineering, especially RMA, security, safety
- Key Performance Parameters (KPPs)
- Open system architecture

Determine realistic, independent baseline software estimates

- Size, effort, cost and schedule
- COTS, reuse and newly developed
- Tasks not reflected in cost models
- Include COTS refresh through both development and sustainment



Software Acquisition Best Practices for Obtaining Contractual Insight

Require key software technical & management deliverables

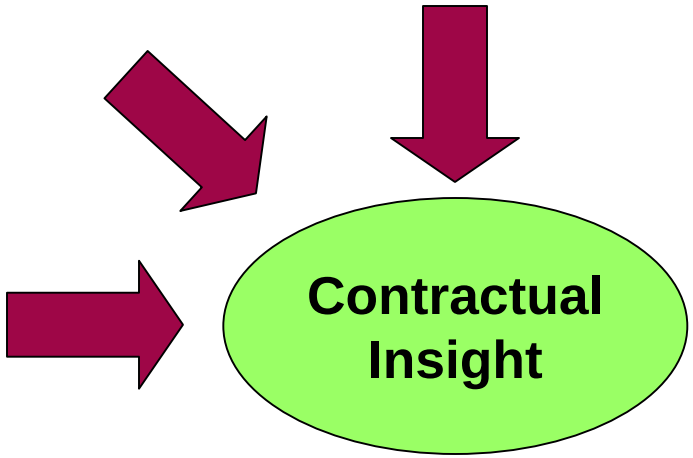
- Highest risk reduction potential:
  - Plans (development, build, transition)
  - Requirements & Architecture
  - Test plans, procedures & reports
  - Metrics reports
  - Delivery, installation, O&M documentation
- Use electronic delivery

Require timely electronic access to all software products

- COTS Evaluation Trade Studies
- Intermediate and Final Products
  - Requirements, Architecture, Design
  - Implementation (including code)
  - Integration & Verification Testing

Require software level technical & management reviews

- In addition to system reviews
- Include COTS software experts in reviews



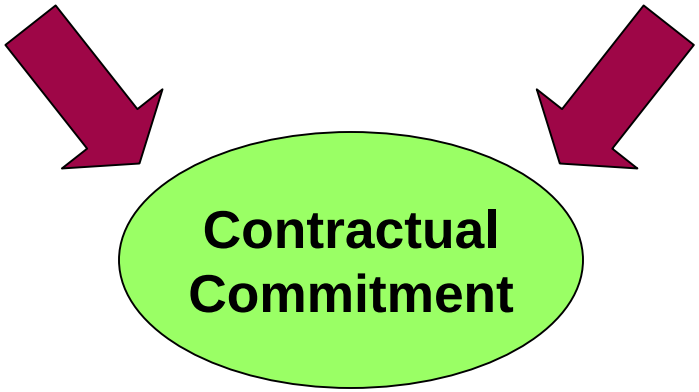
Software Acquisition Best Practices for Obtaining Contractual Commitment

Mandate compliance with robust commercial standard

- For example, EIA/IEEE J-STD-016
- Tailor standard for CBSS development

Require contractor commitment to Software Development Plan

- Require SDP to include processes for COTS software
- Require Integrated Management Plan (IMP) to have adequate systems engineering & sustainment for COTS
- Include commitment to SDP in IMP



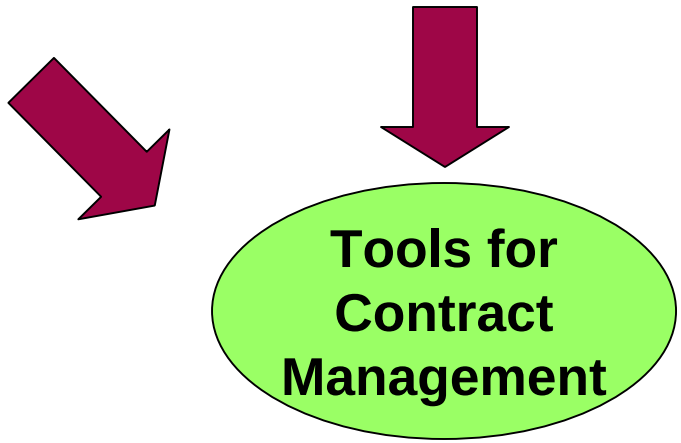
Software Acquisition Best Practices for Providing Tools for Contract Management

Incentivize software quality,* not just cost and schedule

- Use award and incentive fee plans
- Reward adherence to
  - Defined software processes
  - Software process improvement
- Reward timely and adequate response to Government comments
- Reward low rework rates
- Reward meeting performance requirements (e.g., RMA) post delivery/launch
- Reward architecture development that supports CBSS evolution

Mandate periodic team software capability appraisals

- Relate results and improvement actions directly to award fee
- Explicitly include COTS processes in appraisals



* Quality in this context is producing work products that do not require rework in successor activities.

Software Acquisition Best Practices for Selecting a Capable Software Contractor Team

Evaluate software capability/ processes of offeror teams

- Individual team member evaluation insufficient
- Evaluate software capability as a separate subfactor under the Mission Capability factor
- Weight according to software risk

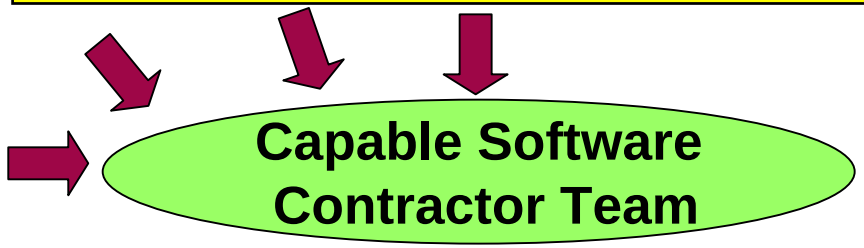
Evaluate realism of cost and schedule bids

- Suspect extremes of productivity, COTS, reuse, low lines of code and short integration times
- Ensure all COTS software tasks are included in the cost & schedule bid
- Ensure bids contain sufficient cost and schedule margin

Evaluate teams' proposed software & related processes

- Corporate and past project process evaluation insufficient
- Include COTS software, systems engineering & logistics processes

Evaluate software and hardware architecture with system design



Software Acquisition Best Practices for Performing Technical Product Review

Focus technical review resources on areas of highest risk

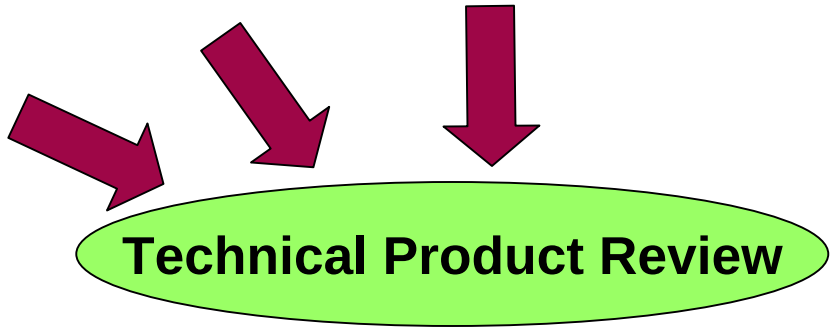
- IPTs, TIMs, working groups, peer reviews, etc.
- Software Level Technical Reviews
- High risk/critical software products (including COTS software)
- Key software technical deliverables

Include users/operators in all technical review activities

- Ensure users/operators understand the evolving CBSS design, including the COTS software capabilities and impacts on O&M

Monitor software integration and verification adequacy

- Begin at the build level
- Focus on areas of highest risk
- Focus on early performance analysis results and meeting KPPs
  - Ensure COTS software performance is measured
- Ensure requirements allocated to COTS software are verified



Software Acquisition Best Practices for Performing Software Process Review

Review effectiveness of team's defined software & related processes

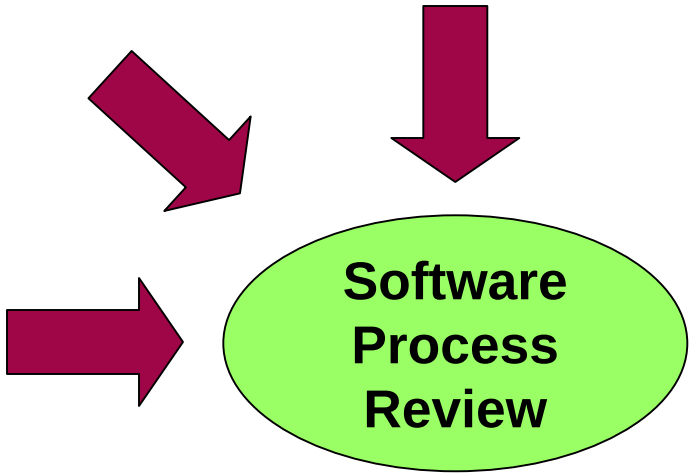
- Identify process deficiencies
  - Especially across team boundaries & with COTS products
- Assist with process improvement
- Individual level 2 & 3 CMMI®/CMM® compliance may not be sufficient

Review team's adherence to defined software & related processes

- Identify adherence deficiencies
- Assist in deficiency correction

Perform periodic team software capability appraisals

- During contract performance
- Support for significant program or award fee milestones
- Explicitly include COTS processes



Software Acquisition Best Practices for Managing the Contract

Use incentive/award fees aggressively

- Motivate good software & related practices
- Focus on quality and architecture

Ensure adherence to software – inclusive requirements

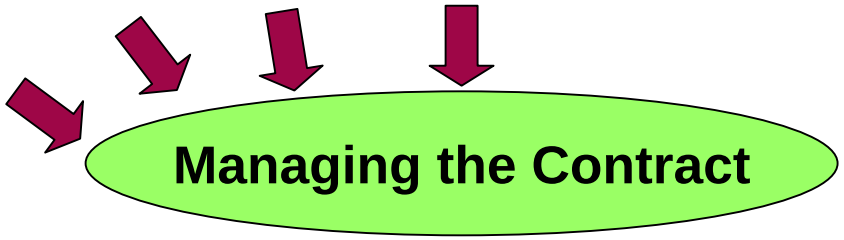
- Especially RMA, safety, & security
- Especially COTS s/w supportability

Apply proactive quantitative management

- Ensure a comprehensive software/ system metrics program balanced across information categories
  - Include leading quality indicators (e.g., rework)
- Perform cross-metric analysis
- Earned value alone is insufficient

Perform periodic independent assessments

- Support for significant program or award fee milestones
- Act aggressively on findings



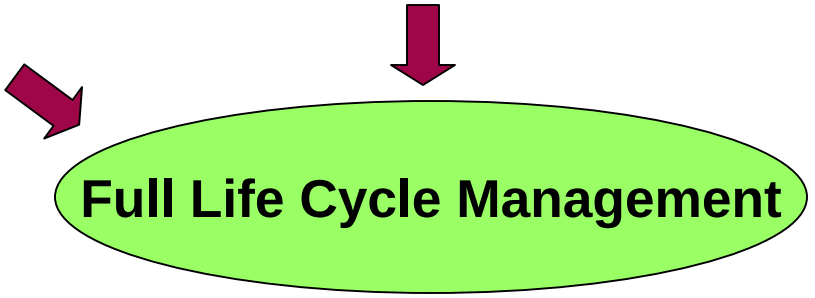
Software Acquisition Best Practices that Span the Life Cycle

Software Acquisition Risk Management

- Continuous software acquisition risk management across all acquisition organization levels
- Program level risk management and contractor development risk management are necessary but not sufficient
- Establish management reserves consistent with software risks (especially COTS software risks)

Software Systems Acquisition

- Integrate software acquisition with the system acquisition process
  - From mission needs identification through system retirement
  - Especially during pre-contract activities



Some CBSS Acquisition Do's and Don'ts

- **DON'T require the following: "Maximize the use of COTS software."**
  - ❖ **DO** require a balanced solution among newly developed, reuse and COTS software that meets cost, schedule and performance objectives.
- **DON'T force the developers to commit to their COTS software selections before they have had time to do thorough evaluations.**
  - ❖ **DO** allow flexibility in schedules for software reviews and deliverables.
- **DON'T force the use of a life cycle model that must define all software requirements up front.**
  - ❖ **DO** provide flexibility to use evolutionary or spiral life cycle models.
- **DON'T blame the contractors for COTS-software related events that are beyond their control.**
  - ❖ **DO** reward mitigating the effects of unforeseen COTS problems.
- **DON'T use commercial item procurements for large, complex ground or space systems.**
  - ❖ **DO** use contracts that require full application of systems and software engineering disciplines.

Conclusion

- **A successful software development project is dependent on the software engineering processes used.**
  - ❖ "The quality of a software product is largely determined by the quality of the process used to develop and maintain it."*
- **However, the software acquisition processes are also highly influential in achieving a successful software development project.**
  - ❖ The software acquisition processes used can positively encourage, or adversely constrain, the developers in their application of high quality software engineering processes.
- **In particular, following "CBSS-friendly" software acquisition best practices will help reduce risk in the acquisition of COTS-based software-intensive systems.**

*** Paulk, M., et al, *The Capability Maturity Model for Software: Guidelines for Improving the Software Process,* Addison-Wesley, 1994, p.8.**

Backup Charts

Space Domain CBSS Lessons Learned - 1

Lesson #1: Critical aspects of CBSS development and sustainment are out of the control of the customer, developer and user.

- Vendors are market driven, and the military is not the market.
- Vendors' strategies and market position may change.
- Product release quality, content and schedules are unpredictable.
- Product and service costs are market driven.

Lesson #2: Full application of system and software engineering is required throughout the CBSS life cycle.

- Using COTS software only shortens part of the software life cycle.
- The CBSS architecture must support COTS software evolution/replacement.
- Hands-on prototyping in a system context is essential.
- Safety, security and supportability must be designed into the CBSS.
- Periodic evaluation of COTS software products using robust evaluation criteria is required.

Space Domain CBSS Lessons Learned - 2

Lesson #3: CBSS development & sustainment require a close, active & continuous partnership among the customer, developer & user.

- Be prepared to trade cost, schedule, performance and O&M concepts.
- Understand which requirements can be relaxed to achieve a COTS-based solution.
- Be active partners to ensure adequacy of major trade decisions and acceptability of the delivered CBSS.

Lesson #4: Every CBSS requires continuous evolution throughout development and sustainment.

- Currency with COTS software upgrades is essential.
- Interfacing external organizations or systems can drive COTS software upgrades, replacements or additions.
- COTS software may need to be replaced or added at any time.
- Modifying COTS software should be a last resort.

Space Domain CBSS Lessons Learned - 3

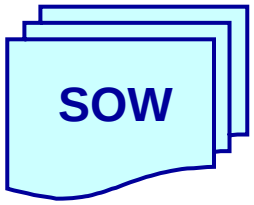
Lesson #5: Current processes must be adapted for CBSS acquisition, development and sustainment.

- Adapt development system and software engineering processes and methodologies to account for COTS software use.
- Reallocate time and effort across the development life cycle.
- Adapt customer and user processes to account for COTS software use.
- Establish standardized licensing, safety certification & security accreditation processes.

Lesson #6: Actual cost and schedule savings with CBSS development and sustainment are overstated.

- Cost and schedule estimates must account for overlooked or underestimated tasks.
- Cost and schedule estimates must account for unexpected impacts (i.e., must contain sufficient margin).

Software Acquisition "Best Practice" Contract



- Comply with SDP
- Do COTS s/w evaluations
- Hold s/w technical reviews
- Undergo periodic software process appraisals

- Software-inclusive, prioritized system requirements
- COTS software support requirements

- Software plans
- Reqs & architecture
- Test documentation
- Metrics reports
- O&M documentation

- Software development process standard (tailored for COTS software)
- Adequate systems engineering & sustainment processes (tailored for COTS)
- All included in IMP as compliant

- Electronic access to all software products
- Access to prime & subcontractor software technical & management data

- Software quality incentives
- COTS architecture evolution incentives

Author Contact Information

- **Richard J. Adams**
  - Senior Engineering Specialist
  - Software Engineering Subdivision, The Aerospace Corporation
  - (310) 336-2907
  - Richard.J.Adams@aero.org

- **Suellen Eslinger**
  - Distinguished Engineer
  - Software Engineering Subdivision, The Aerospace Corporation
  - (310) 336-2906
  - Suellen.Eslinger@aero.org

References - 1

- **Adams, R. J., and S. Eslinger, "COTS-Based Systems: Lessons Learned From Experiences with COTS Software Use on Space Systems," *in Software Technology Conference 2001 Proceedings*, USAF Software Technology Support Center, May 2001.**

- **Adams, R.J., and S. Eslinger, "Lessons Learned From Using COTS Software on Space Systems," *CrossTalk*, 14, 6, 25-30 (June 2001).**
  - http://www.stsc.hill.af.mil/crosstalk/2001/06/adams.html

- **S. Eslinger, "Software Acquisition Best Practices: Experiences from the Space Systems Domain," 2nd OSD Conference on the Acquisition of Software-Intensive Systems, January 2003.**
  - http://www.sei.cmu.edu/products/events/acquisition/

References - 2

- **Adams, R. J. and S. Eslinger, "COTS-Based Systems: Lessons Learned From Experiences with COTS Software Use on Space Systems," Southern California SPIN, 3 October 2003.**
  - ❖ Includes extensive list of COTS software evaluation criteria
  - ❖ http://www.uces.csulb.edu/meetings.html

- **Adams, R. J., S. Eslinger, K. L. Owens, and M. A. Rich, "Reducing Risk in the Acquisition of Software-Intensive Systems: Best Practices from the Space Systems Domain," Space Systems Engineering and Risk Management Symposium Proceedings, February 2004**

Acronyms - 1

Acq	Acquisition
CBSS	COTS-Based Software System
CMM®	Capability Maturity Model®
CMMI®	Capability Maturity Model® IntegrationSM
COTS	Commercial Off the Shelf
DoD	Department of Defense
EIA	Electronic Industries Alliance
Eng	Engineering
H/W	Hardware
IEEE	Institute of Electrical and Electronics Engineers
IMP	Integrated Management Plan
IPT	Integrated Product Team
J	Joint
KPP	Key Performance Parameter

Acronyms - 2

MOIE	Mission-Oriented Investigation and Experimentation
NRO	National Reconnaissance Office
O&M	Operations and Maintenance
OSD	Office of the Secretary of Defense
RFP	Request for Proposal
RMA	Reliability, Maintainability, Availability
S/W	Software
SDP	Software Development Plan
SLOC	Source Lines of Code
SMC	Space and Missile Systems Center
SPIN	Software Process Improvement Network
STD	Standard
TIM	Technical Interchange Meeting
USAF	United States Air Force