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

- **Reviewers**
  - Linda A. Abelson, Software Acquisition and Process Office
  - Peter Hantos, Software Acquisition and Process Office
  - John Cantrell, Software Architecture and Engineering Department

- **Sponsor and Reviewer**
  - Michael Zambrana, USAF Space and Missile Systems Center, Directorate of Systems Engineering

- **Funding Source**
  - Mission-Oriented Investigative Experimentation (MOIE) Research Program (Software Acquisition Task)
Outline

- Background and Definitions
- Scope
  - Software Acquisition Best Practices 2003 Reviewed
  - Scope of Software Acquisition Best Practices 2004
- Software Acquisition Best Practices 2004
  - Early Acquisition Life Cycle Phases
  - Evolutionary Acquisition
- Conclusion
• **Definition:** **Best Practices** are practices that people with recognized expertise in the subject area have identified through experience as being significant contributors to project success.

• **Negative experience or positive experience may identify Best Practices**
  ❖ However, one must not be trapped by logical fallacies

• **Note that Best Practices (both individually and collectively)**
  ❖ Have not necessarily undergone detailed study
  ❖ Have almost never been analytically determined to be “best”
  ❖ Never form an exhaustive set (There is always the possibility of more)
  ❖ Are not static (They change with new experiences and new technologies)
  ❖ Are dependent on the context and environment
Software Acquisition (SA) Best Practices

- **Software Acquisition (SA) Best Practices** are, therefore, practices that people with recognized software acquisition expertise have identified through experience as being significant contributors to the successful acquisition of software-intensive systems.

- The SA Best Practices presented derive from the research team’s collective experience in the acquisition of software-intensive space systems:
  - Over 60 collective years of software acquisition experience spanning approximately 20 years duration
  - Many additional years of experience in developing software, managing software development projects, and leading software process improvement efforts.
Characteristics of Space Systems (SS)

- 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
- Developed by large teams of multiple contractors

*Space Systems Software Acquisition Best Practices must support these characteristics.*
Outline

• Background and Definitions

• Best Practice Scope
  ❖ Software Acquisition Best Practices 2003 Reviewed
  ❖ Scope of Software Acquisition Best Practices 2004

• Software Acquisition Best Practices 2004
  ❖ Early Acquisition Life Cycle Phases
  ❖ Evolutionary Acquisition

• Conclusion
SS SA Best Practice Scope

• Single system development contract for a software-intensive system
• Pre- and post-contract award software acquisition activities for the system development contract
• Full life cycle software acquisition activities spanning the contract award boundary
  ❖ Software Risk Management
  ❖ Software Systems Acquisition
SS SA Best Practices for a System Development Contract

Software Acquisition Domain

- Establish Program Baseline
- Obtain Contractual Insight
- Obtain Contractual Commitment
- Select Capable Contractor Team
- Provide Contract Management Tools

- Perform Technical Product Reviews
- Perform Software Process Reviews
- Manage the Development Contract

Pre Contract			RFP			Development Contract
Proposal			Post Contract

Software Engineering Domain

Contractor
SS SA Best Practice Scope

- Software acquisition activities for the full DoD and National Security Space (NSS) acquisition life cycle
- Pre- and post-contract award software acquisition activities for early DoD and NSS life cycle phases
- Evolutionary acquisition
DoD and NSS Acquisition Models*

NSS Space Acq Policy 03-1

Key Decision Points:

- PHASE A Approval
- PHASE B Approval
- PHASE C Approval

Pre KDP-A Activities

PHASE A (Study Phase) Concept/Architecture Dev

PHASE B (Design Phase) Risk Reduction & Design Development

PHASE C (Build, Test, Launch) Acquisition & Operations Support

Pre-Systems Acquisition

SRR SDR PDR CDR

Systems Acquisition

Concept Refinement

Technology Development

System Development & Demonstration

Production and Deployment

Operations and Support

Sustainment

DoDI 5000.2 (12 May 2003)

Milestones:

- Technology Development Approval
- System Development & Demonstration Approval
- Design Readiness Review
- Low-Rate Initial Prod Approval
- Full Rate Production Approval

* From National Security Space Acquisition Policy #03-01, 6 October 2003.
Outline

• Background and Definitions
• Scope
  ❖ Software Acquisition Best Practices 2003 Reviewed
  ❖ Scope of Software Acquisition Best Practices 2004
• Software Acquisition Best Practices 2004
  ❖ Early Acquisition Life Cycle Phases
  ❖ Evolutionary Acquisition
• Conclusion
Concept Refinement Best Practices
(Pre KDP-A Activities)

Defining:
- Program life cycle
- Initial Government architecture concepts
- Initial Government cost and schedule baselines
- Executable program evolutions
- Global acquisition strategy

NSS Space Acq Policy 03-1
Pre-Systems Acquisition

Key Decision Points:
1. PHASE A Approval

Pre KDP-A Activities

Concept Refinement

Milestones:
1. Technology Development Approval

DoDI 5000.2 (12 May 2003)
Best Practices for Defining the Program Life Cycle

**Use a software-friendly acquisition model**

- Evolutionary acquisition is more suited to large, complex software-intensive systems, such as space systems

**Tailor the acquisition model for software-intensive system**

- SDR level of maturity before MS B/KDP-B
- Selection of a single contractor at appropriate point in software development life cycle
- With or without production phase

**Choose software-friendly points in the life cycle for contract actions**

- Avoid contract actions in the middle of software development spirals (e.g., post System PDR)
- Develop firm basis for software costing before MS B/KDP-B

Program Life Cycle
### Example DoD and NSS Acquisition Models
Tailored for Software-Intensive Systems without Production

**NSS Space Acq Policy 03-1 (Adapted)**

<table>
<thead>
<tr>
<th>Key Decision Points:</th>
<th>Pre-Systems Acquisition</th>
<th>Systems Acquisition</th>
<th>Sustainment</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHASE A/B Approval</td>
<td>PHASE B/C Approval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre KDP-A Activities</td>
<td>PHASE A/B Concept/Architecture Dev &amp; System Design</td>
<td>PHASE B/C Design, Development (Build, Test, Launch) &amp; Operations Support</td>
<td>IOC</td>
</tr>
<tr>
<td>SRR SDR</td>
<td>PDR CDR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concept Refinement</td>
<td>Technology Development &amp; System Design</td>
<td>System Development &amp; Demonstration</td>
<td></td>
</tr>
<tr>
<td>Technology Development &amp; System Design</td>
<td>System Development &amp; Demonstration Approval</td>
<td>Design Readiness Review</td>
<td>Limited Deployment Approval</td>
</tr>
<tr>
<td>A B C</td>
<td>SRR SDR</td>
<td>PDR CDR</td>
<td>Deployment Operations and Support</td>
</tr>
<tr>
<td>Technology Development Approval</td>
<td>System Development &amp; Demonstration Approval</td>
<td>Design Readiness Review</td>
<td>Limited Deployment Approval</td>
</tr>
<tr>
<td>DoDI 5000.2 (12 May 2003) (Adapted)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Best Practices for Developing the Initial Government Architecture Concepts

### Perform software-inclusive architecture trade studies
- With system architecture trades
- Identify and address critical HW/SW architecture issues
- Include major legacy components and COTS software

### Include software in evaluation of architecture concepts
- Evaluate software evolution and growth capability
- Include software in life cycle cost analysis (COTS software refresh, legacy and new software re-engineering and maintenance)

### Select a set of integrated HW/SW architecture concepts
- Able to grow with each successive evolution with little expected rework
- Able to integrate each successive evolution with previous evolutions (and legacy system, as applicable)

---

**Government Architecture Concepts**

---

*THE AEROSPACE CORPORATION*
Best Practices for Developing the Initial Government Cost and Schedule Baseline

Determine **realistic SW size estimates** for each evolution

- Use Gov’t. HW/SW architecture concept
- Include all SW functionality and infrastructure needed
- Use historical data from similar past programs & early concept study data

Determine **realistic SW effort & cost estimates** for each evolution

- Include COTS, reuse and newly developed software
- Include tasks not reflected in cost models (e.g., integration of SW components costed separately, COTS)

Determine **realistic SW schedule estimates** for each evolution

- Include all software effort in schedule
- Never compress software schedule >20% off nominal*

---

Consider **SW implications when defining evolution capabilities**

- Analyze feasibility of developing the required software for each evolution
  - Based on realistic software size, effort, cost and schedule estimates
  - Include software cost and schedule estimation risk
- Analyze feasibility of integrating the software in each evolution with all previous evolutions (and legacy system(s), as applicable)
  - Based on integrated hardware/software architecture
  - Analyze impacts of COTS software refresh and legacy software upgrades

Consider **SW implications when defining evolution schedules**

- Analyze feasibility of overlapping software development schedules for closely spaced evolutions
- Avoid plans that require developing subsequent evolutions on unknown software baselines
- Analyze feasibility of COTS refresh and legacy SW upgrade schedules
Best Practices for Developing the Global Acquisition Strategy

Develop plans for computer system technology insertion

- Include COTS HW and SW refresh in each successive evolution
- Understand new computer HW & SW technologies needed for each evolution and study their readiness

Develop plans for evaluation of contractor software capability

- Perform a Government evaluation of contractor team software capability
- Prior to or part of selection of a single development contractor

Develop plans for software support

- Plan for managing multiple baselines (operations and development)
- Plan for integrating software maintenance actions on operational evolutions into evolutions under development

Global Acquisition Strategy
Principal objective of Phase A/B contract(s)* is to develop the information needed for the Government to:

- Solidify the program definition to establish an executable program
- Update the global acquisition strategy, including acquisition plans and products for this and all future evolutions

* Space systems usually have multiple parallel contracts in this phase, with selection of a single development contractor in the next phase (B/C).
SS SA Best Practices for a Phase A/B Contract

Software Acquisition Domain

- Establish Requirements Baseline
- Develop System Architecture Concept
- Reduce Software Risk

Manage the Phase A/B Contract

Pre Contract

RFP
Proposal

Phase A/B Contract

Post Contract

GOVERNMENT

Contractor

Software Engineering Domain
Include software in Gov’t. system performance requirements

- Specialty engineering, especially RMA
- Key Performance Parameters
- Open system architecture
- Design for evolution and growth

Contract for delivery of SW-inclusive reqs. specifications

- Require System and Segment Specifications as CDRL items
- Use System/Subsystem Specification DID (DI-IPSC-81431a)
Best Practices for Developing the System Architectural Design

Contract for software architecture trade studies

- With system architecture trades
- Include major software legacy components and COTS software

Contract for delivery of system architecture

- Require system architecture as a CDRL item
- Require an integrated HW/SW architecture, defined by multiple architecture views
- Include newly developed, reuse and COTS software
Best Practices for Reducing Software Development Risk

Contract for **software product risk reduction**

- Studies/prototyping of high risk areas for software, e.g.
  - Mission processing algorithms
  - Mission planning concepts
- Simulation development
- Increase readiness level of computer HW and SW technologies

Contract for **software process risk reduction**

- Require delivery of Software Development Plan (DID DI-IPSC-81427a)
- Require compliance with robust software development standard
- Enable contractor team to prepare for software capability evaluation

SW Development Risk Reduction
Best Practices for Managing the Phase A/B Contract

Ensure contractor(s) define software-inclusive reqs. specs.

- Software systems engineers (contractor and Government) must participate with contractor and Gov’t. systems engineers

Ensure contractor(s) define integrated HW/SW architecture

- Software systems engineers (contractor and Government) must participate with contractor and Gov’t. systems engineers

Participate with contractor in software risk reduction

- Government software acquisition personnel with technical expertise in software product and process engineering must participate

Managing the Contract
Evolutionary Acquisition Strategy - 1

Acquisition Planning

Global Acquisition Strategy

Feedback

Pre A

A/B

B/C

Increment 1

B/C

(O&S)

A/B

B/C

Increment 2

(O&S)

A/B

B/C

Increment 3

(O&S)

Feedback

Ongoing or near term

Future planning
Best Practices for Updating the Global Acquisition Strategy

Update **SW-inclusive program baseline**
- Software-inclusive system requirements
- Integrated HW/SW architecture
- Realistic software size, effort, cost & schedule estimates for each evolution

Update **definition of SW-friendly evolutions**
- Evolution capabilities, schedules and integration strategies
- COTS software refresh and legacy software upgrades

Update **software-specific plans**
- Software support strategy
- Contractor team software capability evaluations
- Software technology insertion
- Software transition to operations

Updated Global Acquisition Strategy
Best Practices that Span the DoD and NSS Acquisition Life Cycle

Software Acquisition Risk Management

• **Integrate** software acquisition with the system acquisition process
  • From capability needs identification through system retirement
  • Especially during early acquisition life cycle phases

Software Systems Acquisition

• **Continuous software acquisition risk management**
  • Across the entire acquisition life cycle
  • Across all evolutions
  • Within each ongoing evolution
• Program level risk management and contractor development risk management are necessary but not sufficient
Outline

• Background and Definitions
• Scope
  ❖ Software Acquisition Best Practices 2003 Reviewed
  ❖ Scope of Software Acquisition Best Practices 2004
• Software Acquisition Best Practices 2004
  ❖ Early Acquisition Life Cycle Phases
  ❖ Evolutionary Acquisition
• Conclusion
Conclusion

• Software acquisition best practices do not guarantee success
  ❖ They are not a panacea!

• Using best practices, however, can reduce risk in complex software-intensive system acquisitions

• Evolutionary acquisition, in particular, is a complex strategy that requires careful planning and execution in order to achieve its anticipated benefits

• Software acquisition best practices will be most effectively implemented if done in the context of a software acquisition process improvement program
  ❖ Based on experiences with software development

Section 804 of the FY03 Defense Authorization Act requires the establishment of software acquisition process improvement programs.
Back-Up Charts

- Software Acquisition Best Practices 2003
- Acronym List
- Author Contact Information
Perform **software architecture-inclusive** trade studies

- With system architecture trades
- Include major legacy components
- Supports Government software architecture baseline selection

Determine **realistic, independent** baseline software estimates

- Size, effort, cost and schedule
- COTS, reuse and newly developed
- Tasks not reflected in cost models
- Realism especially critical for evolutionary acquisition

Include **software** in system performance requirements

- Specialty engineering, esp. RMA
- Key Performance Parameters
- Open system architecture

Program Baseline
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 & maintenance documentation
- Use electronic delivery

Require **timely** electronic access to **all** software products

- Requirements
- Architecture, Design
- Implementation (including code)
- Integration and Verification Testing
- Intermediate and Final Products

Require **software level** technical & management reviews

- In addition to system reviews

Contractual Insight
Best Practices for Obtaining Contractual Commitment

Mandate compliance with robust full life cycle SW dev. standard

- For example, EIA/IEEE J-STD-016

Require contractor commitment to Software Development Plan

- Include commitment in Integrated Master Plan (IMP)

Contractual Commitment
Best Practices for Selecting a Capable Software Contractor Team

<table>
<thead>
<tr>
<th>Evaluate software capability as part of source selection</th>
<th>Evaluate software architecture with system design</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Evaluate software capability of offeror <strong>teams</strong></td>
<td>• Evaluate major HW/SW architecture issues (e.g., space-ground trades, reuse of legacy components)</td>
</tr>
<tr>
<td>• Individual team member evaluation insufficient</td>
<td></td>
</tr>
<tr>
<td>• Evaluate software capability/processes as <strong>subfactor</strong></td>
<td></td>
</tr>
<tr>
<td>• Under Mission Capability factor</td>
<td></td>
</tr>
<tr>
<td>• Weight according to software risk</td>
<td></td>
</tr>
<tr>
<td>• Evaluate teams’ <strong>proposed</strong> software processes</td>
<td></td>
</tr>
<tr>
<td>• Corporate and past project process evaluation insufficient</td>
<td></td>
</tr>
</tbody>
</table>

- Evaluate realism of cost and schedule bids
  - • Suspect extremes of productivity, COTS & reuse, & low lines of code

---

**Capable Software Contractor Team**

---

THE AEROSPACE CORPORATION
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 RMA requirements post delivery/launch

Mandate **periodic team** software capability appraisals

- Relate results and improvement actions directly to award fee

* Quality in this context is producing work products that do not require rework in successor activities.
Best Practices for Performing Technical Product Reviews

Perform **in-depth technical reviews** of software products

- IPTs, TIMs, working groups, peer reviews, etc.
- Software Level Technical Reviews
- High risk/critical software products
- Key software technical deliverables
- Focus on areas of highest risk

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

Include **users/operators** in all technical review activities

- Focus on operational suitability of evolving software-intensive system

Technical Product Reviews
Best Practices for Performing Software Process Reviews

Review **effectiveness** of contractor team’s SW processes

- Review team’s adherence to defined software processes
  - Identify adherence deficiencies
  - Assist in deficiency correction
- Evaluate effectiveness of defined SW processes
  - Identify process deficiencies
  - Assist with process improvement
- Level 2 & 3 CMMI®/CMM® adherence for an individual team member may not be sufficient*

Perform **periodic team** software capability appraisals

- During contract performance
- Support for significant program or award fee milestones

* CMM and CMMI are registered trademarks of Carnegie Mellon University.
Best Practices for Managing the Development Contract

**Use incentive/award fees aggressively**
- Motivate good software practices
- Focus on quality

**Ensure satisfaction of software—inclusive requirements**
- Especially RMA

**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

Managing the Contract
# Acronyms and Abbreviations - 1

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acq</td>
<td>Acquisition</td>
</tr>
<tr>
<td>CDR</td>
<td>Critical Design Review</td>
</tr>
<tr>
<td>CDRL</td>
<td>Contract Data Requirements List</td>
</tr>
<tr>
<td>CMM®</td>
<td>Capability Maturity Model®</td>
</tr>
<tr>
<td>CMMI®</td>
<td>Capability Maturity Model® Integration℠</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off the Shelf</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>Dev</td>
<td>Development</td>
</tr>
<tr>
<td>DID</td>
<td>Data Item Description</td>
</tr>
<tr>
<td>DoD</td>
<td>Department of Defense</td>
</tr>
<tr>
<td>DoDI</td>
<td>DoD Instruction</td>
</tr>
<tr>
<td>EIA</td>
<td>Electronic Industries Alliance</td>
</tr>
<tr>
<td>FY</td>
<td>Fiscal Year</td>
</tr>
<tr>
<td>Gov’t.</td>
<td>Government</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>IMP</td>
<td>Integrated Management Plan</td>
</tr>
<tr>
<td>IPT</td>
<td>Integrated Product Team</td>
</tr>
<tr>
<td>IOC</td>
<td>Interim Operational Capability</td>
</tr>
<tr>
<td>J</td>
<td>Joint</td>
</tr>
<tr>
<td>KDP</td>
<td>Key Decision Point</td>
</tr>
<tr>
<td>KPP</td>
<td>Key Performance Parameter</td>
</tr>
<tr>
<td>MOIE</td>
<td>Mission-Oriented Investigation and Experimentation</td>
</tr>
<tr>
<td>MS</td>
<td>Milestone</td>
</tr>
<tr>
<td>NSS</td>
<td>National Security Space</td>
</tr>
<tr>
<td>O&amp;S</td>
<td>Operations and Support</td>
</tr>
<tr>
<td>OSD</td>
<td>Office of the Secretary of Defense</td>
</tr>
<tr>
<td>PDR</td>
<td>Preliminary Design Review</td>
</tr>
<tr>
<td>RFP</td>
<td>Request for Proposal</td>
</tr>
<tr>
<td>RMA</td>
<td>Reliability, Maintainability, Availability</td>
</tr>
<tr>
<td>SA</td>
<td>Software Acquisition</td>
</tr>
<tr>
<td>SDP</td>
<td>Software Development Plan</td>
</tr>
<tr>
<td>SDR</td>
<td>System Design Review</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>SLOC</td>
<td>Source Lines of Code</td>
</tr>
<tr>
<td>SM</td>
<td>Service Mark</td>
</tr>
<tr>
<td>SRR</td>
<td>System Requirements Review</td>
</tr>
<tr>
<td>SS</td>
<td>Space System</td>
</tr>
<tr>
<td>STD</td>
<td>Standard</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>TIM</td>
<td>Technical Interchange Meeting</td>
</tr>
<tr>
<td>USAF</td>
<td>United States Air Force</td>
</tr>
</tbody>
</table>
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

- **Karen L. Owens**
  - Senior Engineering Specialist
  - Software Engineering Subdivision, The Aerospace Corporation
  - (310) 336-5909
  - Karen.L.Owens@aero.org

- **Mary A. Rich**
  - Principal Director
  - Software Engineering Subdivision, The Aerospace Corporation
  - (310) 336-5313
  - Mary.A.Rich@aero.org