Acquisition of Software-Intensive Systems Conference

Software Acquisition Best Practices:
Experiences from the Space Systems Domain

Suellen Eslinger

Software Acquisition and Process Office

Software Engineering Subdivision

The Aerospace Corporation

January 29, 2003

THE AEROSPACE CORPORATION

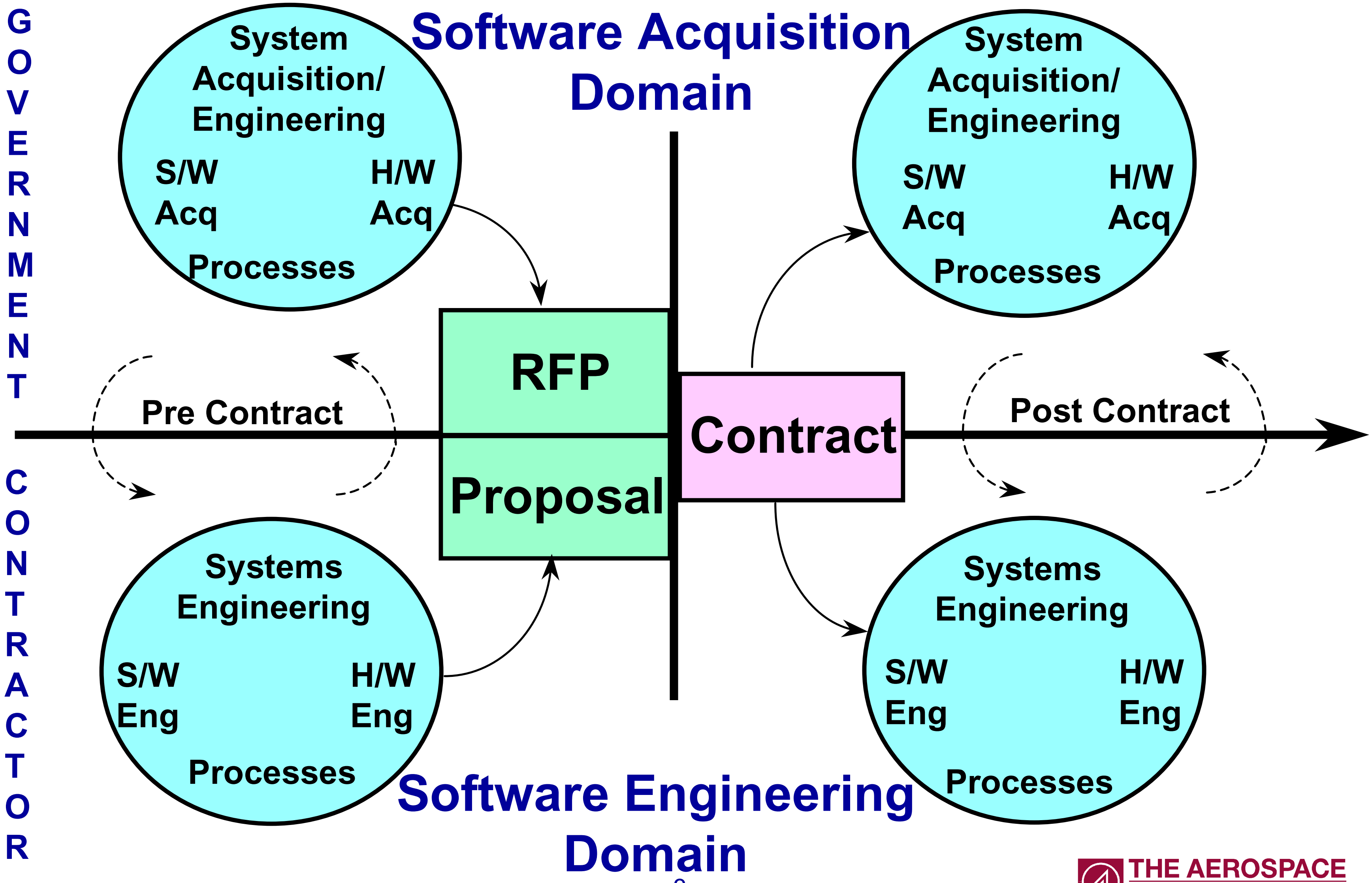
Acknowledgements

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

- **Research co-investigators and reviewers**
  - ❖ Linda A. Abelson, Software Acquisition and Process Office
  - ❖ Richard J. Adams, 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

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

Software Acquisition vs. Software Engineering



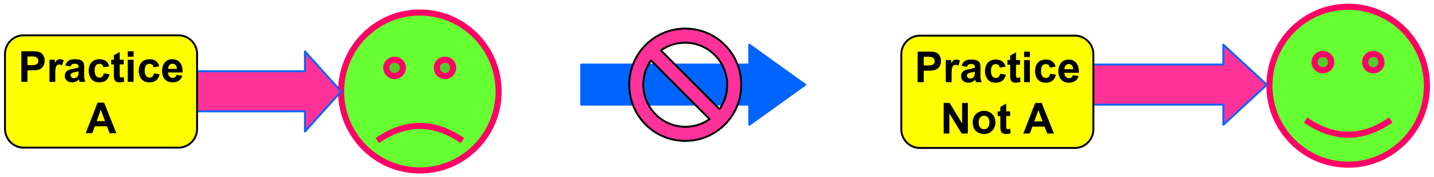
Software Acquisition vs. Software Engineering

- **Software Acquisition and Software Engineering are not the same**
  - ❖ Software Acquisition is the set of processes used by the Government to acquire software
  - ❖ Software Engineering is the set of processes used by the developer to build software
- **Clearly, 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

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

Best Practices

- **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)

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.**

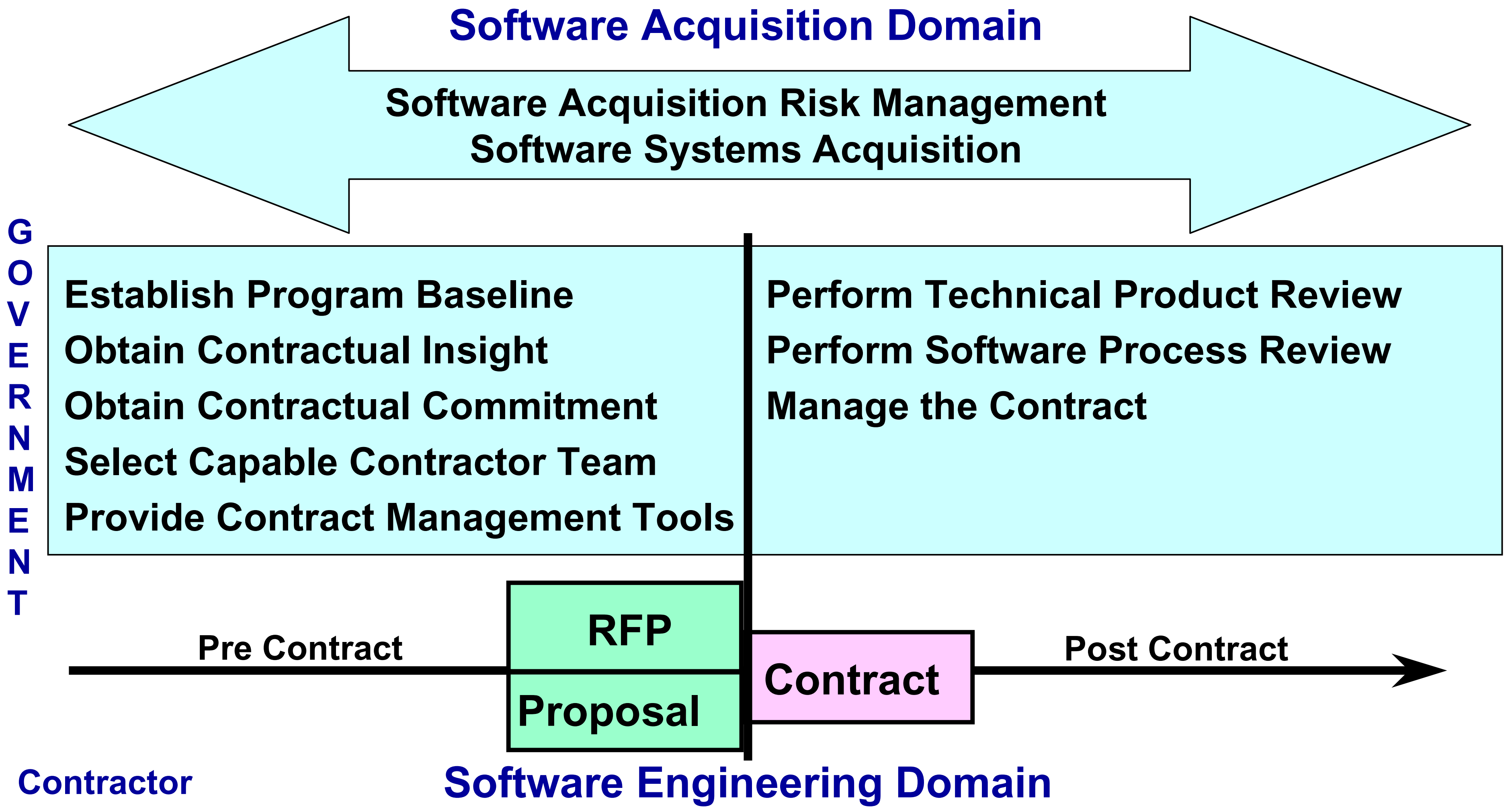
- **The SA Best Practices presented derive from the research team's collective experience in the acquisition of software-intensive space systems**
  - ❖ Over 50 years collective software acquisition experience
  - ❖ Over 18 years duration

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**

Space Systems Software Acquisition Best Practices must support these characteristics.

SS SA Best Practice Roadmap



Software Acquisition Domain

Software Acquisition Risk Management
Software Systems Acquisition

GOVERNMENT

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

Perform Technical Product Review
Perform Software Process Review
Manage the Contract

Pre Contract

RFP

Proposal

Contract

Post Contract

Contractor

Software Engineering Domain

Best Practices for Establishing the Program Baseline

Perform software architecture trade studies

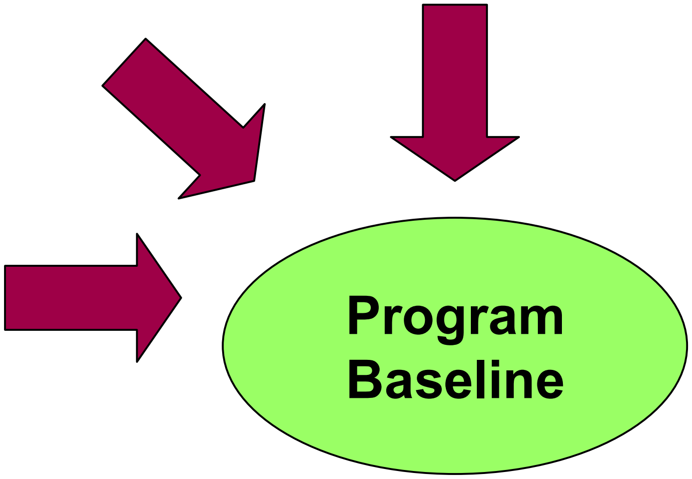
- With system architecture trades
- Include major legacy components
- Supports Government software architecture baseline selection

Include software in system performance requirements

- Specialty engineering, esp. RMA
- Key Performance Parameters
- 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
- Realism especially critical for evolutionary acquisition



Program Baseline

Best Practices for Obtaining Contractual Insight

Require key software technical & management deliverables

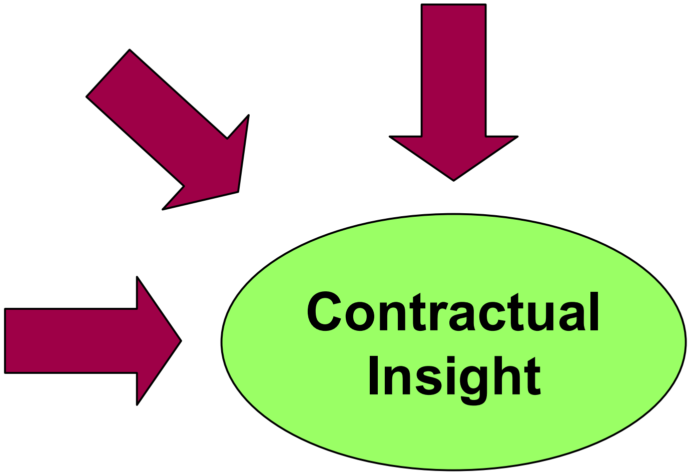
- Highest risk reduction potential:
  - Plans (development, build, transition)
  - Requirements & Architecture
  - Test plans, procedures & reports
  - Metrics reports
  - Delivery, installation & maintenance documentation
- Use electronic delivery

Require timely electronic access to all software products

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

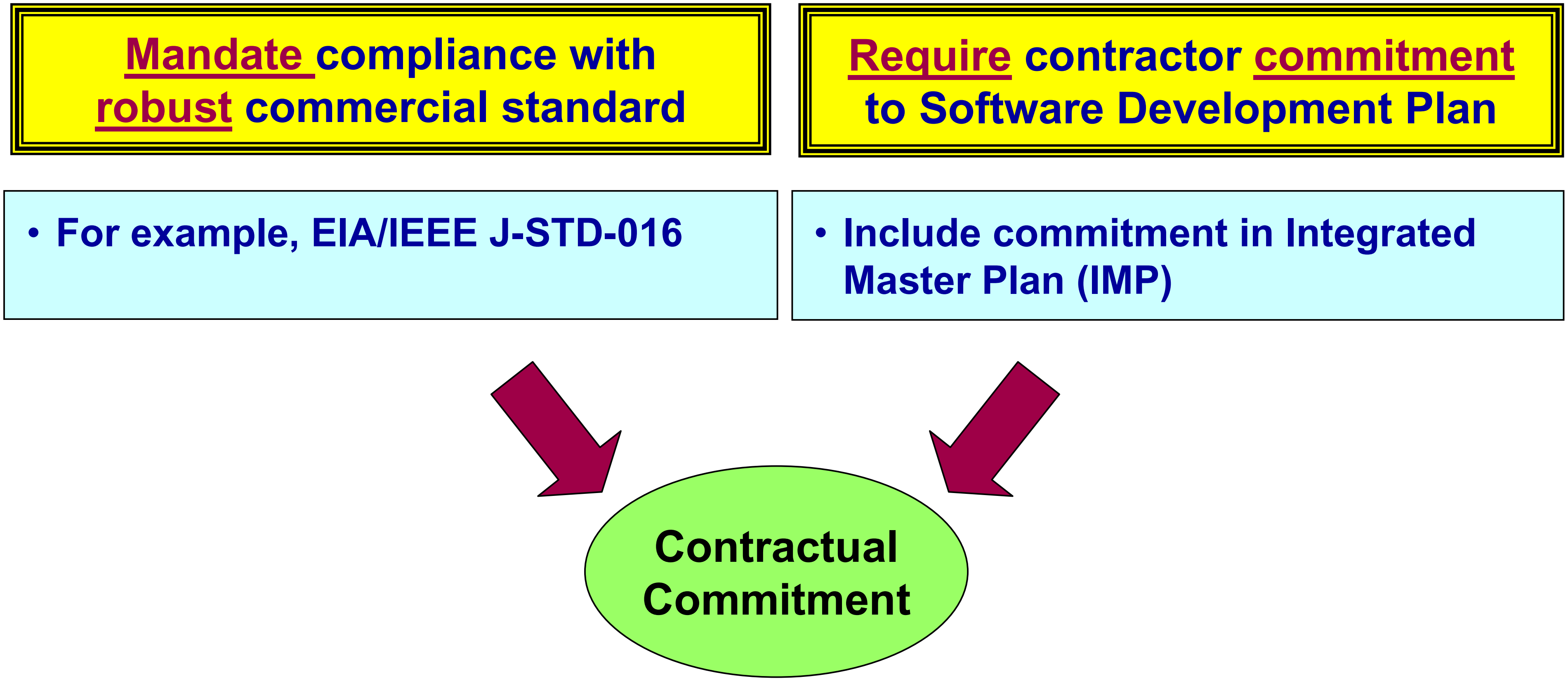
Require software level technical & management reviews

- In addition to system reviews



Contractual Insight

Best Practices for Obtaining Contractual Commitment



Mandate compliance with robust commercial 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

Evaluate software capability of offeror teams

- Individual team member evaluation insufficient

Evaluate teams' proposed software processes

- Corporate and past project process evaluation insufficient

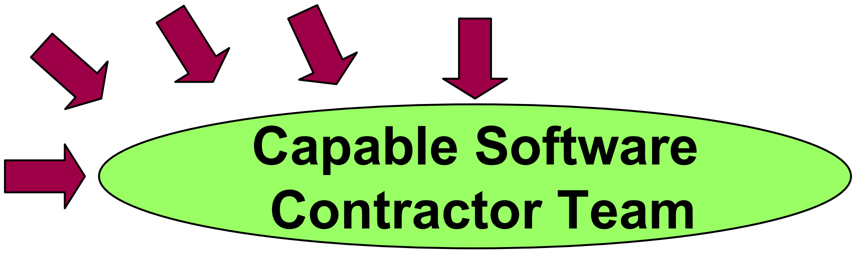
Evaluate software capability/ processes as subfactor

- Under Mission Capability factor
- Weight according to software risk

Evaluate realism of cost and schedule bids

- Suspect extremes of productivity, COTS, reuse and low lines of code

Evaluate software architecture with system design



Capable Software Contractor Team

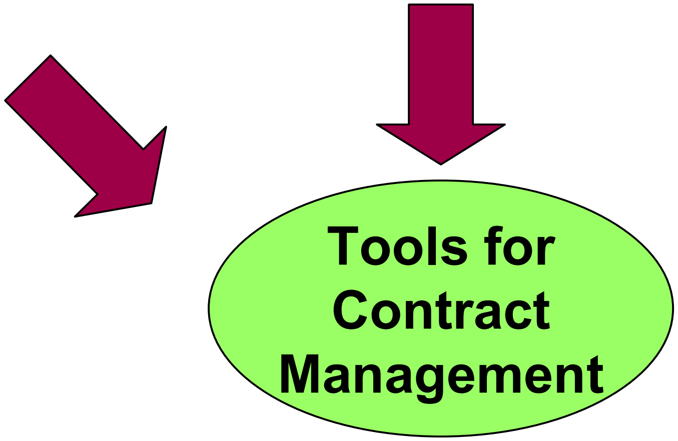
Best Practices for Providing Tools for Contract Management

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 RMA requirements post delivery/launch

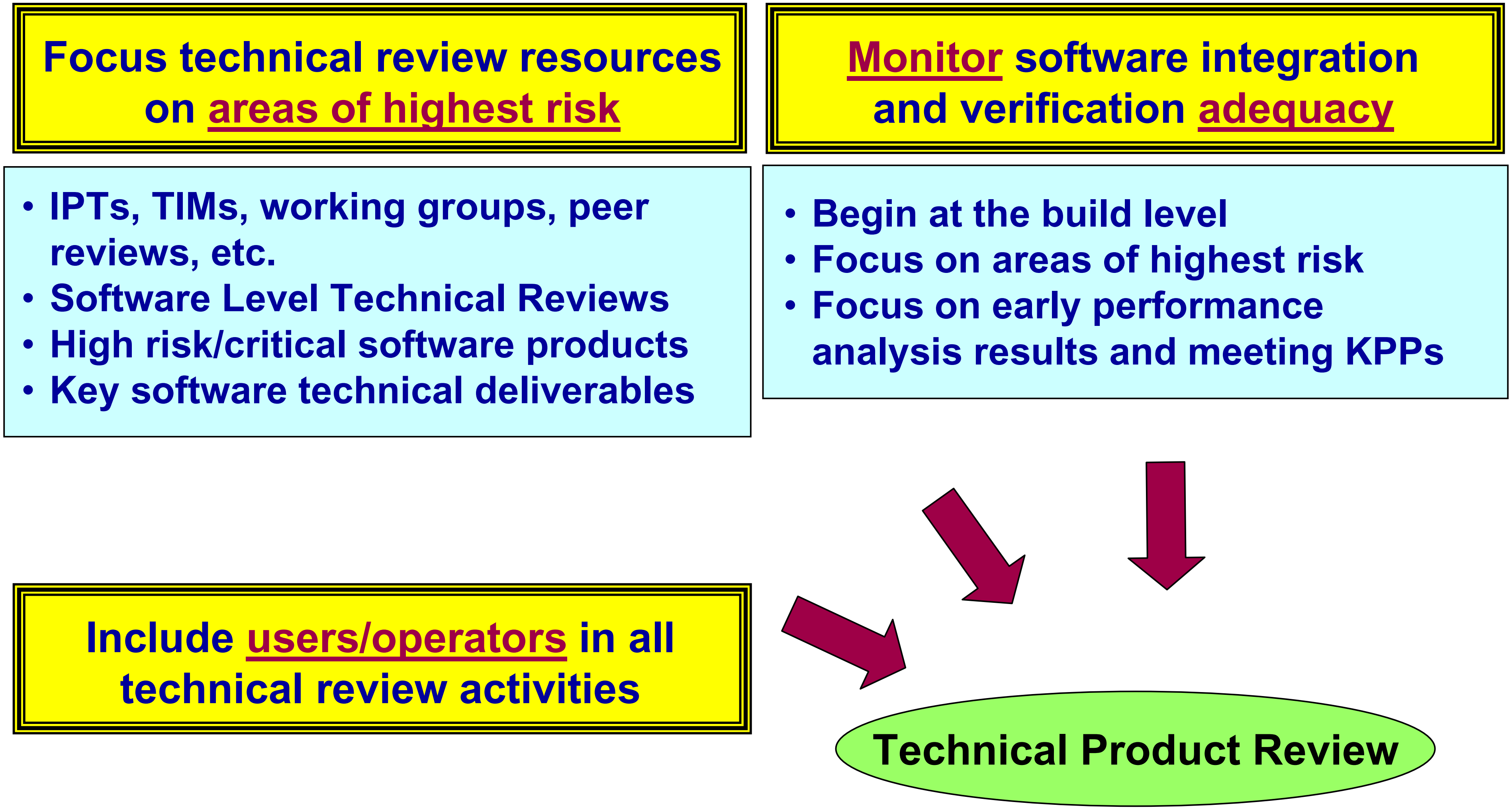
Mandate periodic team software capability appraisals

- Relate results and improvement actions directly to award fee


Tools for Contract Management

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

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
- Key software technical deliverables

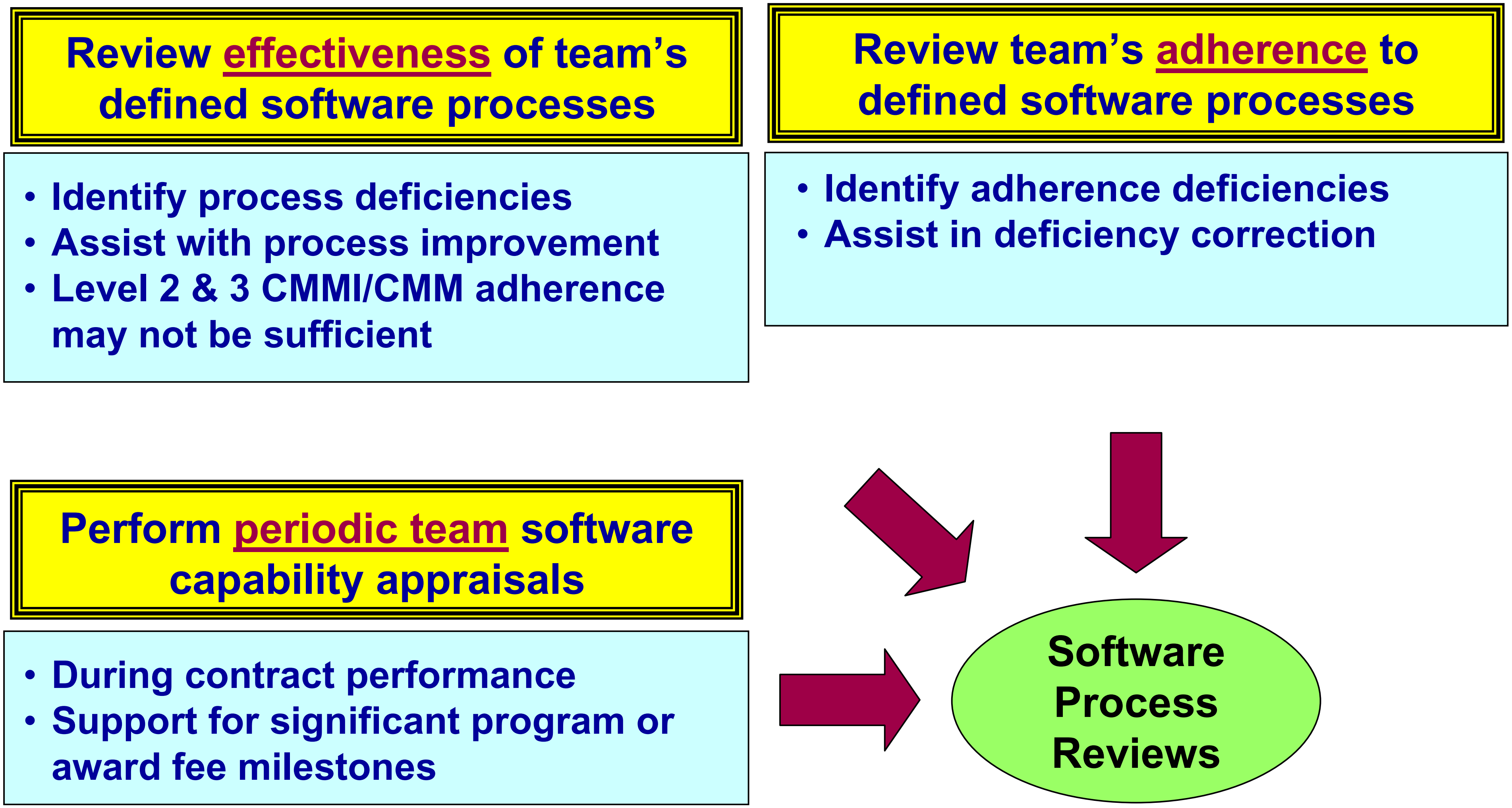
Monitor software integration and verification adequacy

- 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

Technical Product Review

Best Practices for Performing Software Process Review



Review effectiveness of team's defined software processes

- Identify process deficiencies
- Assist with process improvement
- Level 2 & 3 CMMI/CMM adherence may not be sufficient

Review team's adherence to defined software 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

Best Practices for Managing the Contract

Use incentive/award fees aggressively

- Motivate good software practices
- Focus on quality

Ensure adherence to software–inclusive requirements

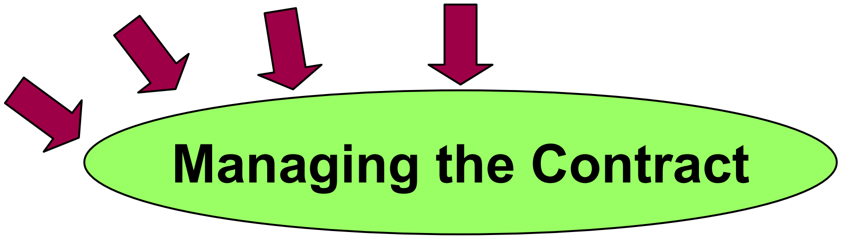
- Especially RMA

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

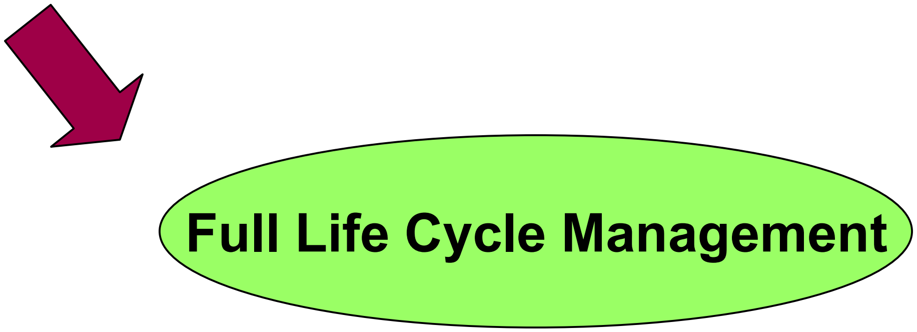
Best Practices that Span the Life Cycle

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

Software Systems Acquisition

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



Full Life Cycle Management

The Way Ahead: Software Acquisition Process Improvement

- **Need to institute a software acquisition process improvement program for the acquisition organization**
  - ❖ Define, document, measure and improve software acquisition processes
  - ❖ Make an integral part of a system acquisition process improvement program
  - ❖ Collect acquisition data consistently across programs to provide a library of historical data
    - – Basis for estimation of future software-intensive system acquisitions
    - – Foundation for analyzing effectiveness of best practices

Ideally, software acquisition process improvement should be model based. Adding software and system acquisition disciplines to the CMMI would be the most effective approach!

Best Practices for Establishing the Program Baseline

- **Perform software architecture trade studies**
  - v Part of system architecture trade studies
  - v Include major legacy components (especially software) likely to be proposed
  - v Use as basis for selecting a Government software architecture baseline
- **Determine realistic, independent Government estimates for the baseline software architecture**
  - v Software size, effort, cost, schedule
  - v Include COTS, reuse and newly developed software
  - v Include effort for tasks not easily estimated by standard cost models
  - v Realistic software size and schedule estimates are especially important for successful evolutionary acquisition
- **Include software in system performance requirements**
  - ❖ **Especially important** for specialty engineering requirements (e.g, reliability, maintainability, availability, safety, security, supportability, testability, fault tolerance)
  - ❖ Key Performance Parameters
  - ❖ Open system architecture requirements

Best Practices for Obtaining Contractual Insight

- **Require software technical and management (electronic) contract deliverables with highest risk reduction potential**
  - v Software plans (Software development plan, master build plan, plan for transition to O&M)
  - v Software and interface requirements specifications
  - v Software architecture descriptions
  - v Software test plans, procedures, reports
  - v Software metrics reports
  - v Delivery, installation and maintenance documentation (Software product specifications, software version descriptions)
- **Require timely electronic access to all intermediate and final software products**
  - v Obtain detailed design and code via electronic access
- **Require software level technical and management reviews**
  - ❖ Software level information provided in system level reviews is not sufficient

Best Practices for Obtaining Contractual Commitment

- Contractually mandate compliance with a robust commercial software process standard
  - v For example, EIA/IEEE J-STD-016
- Require the contractor to contractually commit to following their Software Development Plan
  - v Include this commitment in the contractually compliant Integrated Master Plan (IMP)

Best Practices for Selecting a Capable Software Contractor Team

- **Evaluate the software development capability of the offeror teams**
  - Evaluating only the prime or evaluating each team member independently is not sufficient
- **Evaluate the adequacy of the offeror team's software development processes proposed for use on the program under bid**
  - Evaluating only corporate processes or processes used on past projects is not sufficient
- **Make the evaluation of software development capability/processes a separate subfactor under the Mission Capability factor**
  - Make the weight in the source selection commensurate with software risk
- **Evaluate the realism of the offerors' software cost and schedule bids (Less is not necessarily better!)**
  - Suspect overly optimistic productivity, COTS and reuse and low estimates of new lines of code
  - Ensure adequacy of systems engineering, program management and verification resources related to software
- **Evaluate adequacy of the proposed software architecture as part of the system design evaluation**

Best Practices for Providing Tools for Contract Management

- **Incentivize software quality*, not just cost and schedule (via award and incentive fee plans)**
  - Incentivize adherence to defined software processes and software process improvement
  - Incentivize timeliness and adequacy of responsiveness to Government technical product and process review comments
  - Incentivize low rework rates
  - Incentivize meeting RMA requirements post delivery/launch
- **Contract for periodic contract monitoring software capability appraisals of the contractor team**
  - ❖ 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 Review

- **Focus Government technical review resources on areas of highest risk**
  - Contractor meetings
    - IPTs, TIMs, working groups, peer reviews, etc.
    - Software Level Technical Reviews
  - Technical products
    - High risk/criticality areas of all software products
    - Key software technical deliverables
- **Monitor adequacy of software integration and verification testing, beginning at the software build level**
  - Focus on areas of highest risk
  - Focus on early performance analysis results and meeting KPPs
- **Include the users/operators in all technical review activities**

Best Practices for Performing Software Process Review

- **Review contractor team's defined software processes for <u>effectiveness</u>**
  - ❖ Identify process deficiencies to assist with software process improvement
  - ❖ Level 2 & 3 CMMI/CMM model adherence may not be sufficient
- **Evaluate contractor team's processes for <u>adherence</u> to their defined processes**
  - ❖ Identify process adherence deficiencies
  - ❖ Assist in correcting process adherence deficiencies
- **Perform <u>periodic</u> contract process monitoring software capability appraisals of the contractor *<u>team</u>***
  - ❖ Performed during contract performance per contract provisions
  - ❖ Support for significant program or award fee milestones

Best Practices for Managing the Contract

- **Use award fee aggressively to motivate good software practices**
  - v Focus on quality, not just cost and schedule
- **Apply proactive quantitative management, based on a robust software/system metrics program**
  - v Ensure contractor team’s metrics program
    - v Is comprehensive and balanced across information categories
    - v Contains sufficient leading indicators of quality problems (i.e., downstream rework)
  - v Perform cross-metric analysis of the contractor team’s monthly reported metrics
  - v Managing by earned value alone is not sufficient
- **Ensure contractor team adherence to software-inclusive system performance requirements**
  - v Especially RMA
- **Charter independent assessments at regular intervals/milestones**
  - v Support for significant program or award fee milestones
  - v Take aggressive action based on independent assessment team recommendations

Best Practices That Span the Life Cycle

- **Software Acquisition Risk Management**
  - ❖ Perform **continuous software acquisition risk management** throughout the life cycle at all levels of the program's acquisition organization
  - ❖ Development risk management is necessary but not sufficient
  - ❖ Program level risk management is necessary but not sufficient

- **Software Systems Acquisition**
  - ❖ Include software acquisition as an **integral part** of the system acquisition process from mission needs identification through system retirement
  - ❖ Especially important is the participation of knowledgeable software acquisition personnel in all pre-contract award activities