Some Considerations for Improving the Acquisition Process

Noel Longuemare

August 11, 2005

Topics

- Is there a Root Cause to our Perennial Problem of Program Overruns, Schedule Slips, Performance Surprises?
- Some Symptoms and Issues
- Some Remedies are already in place (Almost)
  - Evolutionary Requirements
  - CAIV
  - Evolutionary Acquisition
  - Technology can help---Model Driven Architecture
- Program Instability
- Joint Programs---a Major Acquisition Problem
- Improving Program Execution
- What can we do about all this?

Is there a Root Cause to our Perennial Problem?

(Overruns, Schedule Slips, Performance Surprises)

- Probably not!
  - The DoD Acquisition Product Mix is very complex
  - There are distinct differences between product sectors (e.g. ships vs. space vs. radios)
  - There are multiple contributing causes
- The sheer scale of the DoD process makes a certain number of Acq failures inevitable
  - [*Monitored Metrics* should be employed to track and focus attention to minimize these, however]
- There are some common factors, however, that if improved, can have a substantial favorable impact.

Some Symptoms and Issues

- Poorly or improperly executed reform concepts have given some key improvements an undeserved bad rap (eg. Spiral Acquisition)
- Improper or Non-Application of CAIV appears prevalent
- Inexperienced Program Management (both Government and Industry) contributes to program problems
- The Acquisition process still tends to be too serial; it is really should be an iterative approach
- Overruns are not only tolerated, they are anticipated and almost expected as standard procedure, with little of no consequences
- Externally caused Program Instability continues and contributes to huge cost and schedule problems
- Joint Programs are especially difficult; despite rhetoric, Service loyalties still prevail, often to the detriment of Joint success
- Lack of a real System-of-Systems (SoS) Engineering process in the DoD is a major shortfall, especially in the move to Netcentric Ops. Resulting sub-optimization and stovepipe architectures costs millions and fails to deliver needed performance

Some Remedies Are Already In Place (Almost!)

- Five Complimentary Thrusts:
  - Evolutionary (Spiral) Requirements---speeding delivery of needed capabilities while accommodating lessons learned, technology maturation, changing needs, etc.
  - CAIV---fitting program goals to the available dollars
  - Evolutionary (Spiral) Acquisition---Disciplined approach for rapidly fielding needed baseline capabilities
  - Employment of the Modular Open Systems Approach (MOSA)---makes Spiral upgrades feasible and affordable
  - Employment of Model Driven Architecture (MDA) for Systems/Software development---Watershed change in time/cost of systems/software design and maintenance

If properly employed and executed, these thrusts, in combination, can make a significant improvement in overall DoD Acquisition

Evolutionary (Spiral) Requirements

- The Capability Based Requirements process is still too serial and time consuming
- It's not clear that a streamlined path exists for CAIV iterations to affect results
- A more direct/formal tie between Requirements Spirals and Acquisition Spirals is needed to benefit from lessons-learned feedback

Requirements generation should be an iterative process---provisions are needed to facilitate this

Cost as an Independent Variable (CAIV)

- Fitting Requirements (and Quantities) to available funds is a fundamental need
- A technically and operationally rigorous process is needed to establish baseline KPPs clearly separated from "desirements" (Goals)
- To function properly, streamlined but realistic cost estimates (sensitive to technology advances) are required
- Where feasible, multiple alternatives should be sought out, preferably through competitive actions
- Where possible, "ACTD-like" demos can provide real-world feedback to add further realism to this process

An Iterative CAIV-driven Requirements Process is key---provisions are needed to facilitate this

Evolutionary (Spiral) Acquisition

- Selecting a properly balanced *Baseline Spiral* is key (i.e., providing a "Block 0" capability in the shortest feasible time that is a useful improvement over current)---Close cooperation with the Requirements Community required
- Extensive use of *Modeling and Simulation* should be enforced as part of the source selection criteria and monitored thereafter
- *Demos* of key functions and capabilities should be a default part of the milestone process (with high level approval required for a waiver)
- Enforcement of a *MOSA* strategy is key to affordable future spirals
- Early involvement of *Users and OT&E* is essential
- *Shared contract incentives* (not FFP contracts) based on cost reductions to the government should be standard fare

More frequent reviews of program performance with reduced tolerance for overruns is needed

Model Driven Architecture (MDA)

- MDA represents a watershed change in the cost, quality, and time to design, test, field and maintain systems and software
- Instead of paper specs, executable models (xUML)are used to specify (and assess) detailed system performance
- System designers construct, evaluate, and optimize models of their design; once satisfied, generation of actual software (C++, etc.) is automated with little human intervention
- Configuration Control is applied to the Models, not software, greatly facilitating change design and implementation
- Although relatively embryonic, sufficient experience now shows this to be practical and feasible. Substantial improvements are to be expected in the future as this matures and use expands
- Industry has recognized these benefits and is rapidly moving to exploit this technology

Orders of magnitude reductions in cost and time to field quality software are now within reach

Some Model Driven Architecture Metrics

[Software development metrics from an independent assessment of Missile Defense Agency's Model Driven Architecture approach for its Automated Battle Management (ABM) development]

- Phase Containment Effectiveness (PCE) – *[tracks ability of each phase to find defects before progressing to next phase]:*
  - Industry average (defects found): 58%
  - Industry using best practices: 70-80%
  - ***ABM (increment 0) using MDA: 97%***

- Defect Density (number of defects per KSLOC)
  - Industry average for new code: 6.1 bugs per KSLOC
  - ***ABM (increment 0): 0.2 bugs per KSLOC***

Conclusion: MDA significantly reduces development cost and schedules

Program Instability

- Historical data consistently shows Program Instability to be a cause of huge cost and schedule overruns
- The basic DoD budgetary and fiscal management processes stimulate actions that aggravate this situation
- Inability for DoD programs to maintain a legitimate program reserve contributes to this and causes a cascading effect that is now considered normal
- Without funds to cover contingencies on a priority program that inevitably arise, DoD budgeteers are forced to destabilize numerous other programs
- The Program Stability Fund Initiative---started in 1998 but since abandoned---is one concrete, workable approach toward alleviating much of this problem.
- Prioritizing and adequately funding fewer projects---coupled with Stability Reserves---can pay great dividends

Reducing Program Instability can free billions and accelerate fielding of key capabilities by years

Joint Programs---a Major Acquisition Problem

- Modern warfare demands an ever-greater degree of seamless Joint Operations in all areas
- Ironically, Joint Programs are the most prone to failure/major problems; conflicting Service-unique priorities are a fundamental impediment
- COCOMS have the real need, but have a limited voice in acquisition priorities and joint program execution
- A 2003 DSB study: "Enabling Joint Force Capabilities -Phase I" addresses this issue and makes workable recommendations on how to address significant aspects of this problem
- The 2005 DSB Summer Study currently underway is making further recommendations in this area
- Although Netcentric Ops involve complex Systems of Systems, the DoD has essentially no Systems Engineering capability to study, synthesize, optimize, and prioritize these complex systems

The severity of these issues coupled with the growing need for more "Jointness" demands that the DoD come to grips with these problems

Improving Program Execution

- The experience level and technical depth of government PMs has decreased in recent years; Industry suffers some of the same problems
- This results in poor decisions, failure to recognize emerging problems early, and overall poor program execution
- DoD needs to systematically improve quality in this area, including greater use of IPAs as well as upgrading organic personnel
- Department wide Metrics on program execution can form the basis for more pointed program reviews and identification of what is working and what isn't, as well as levying of execution goals on Service SAEs, PEOs, etc.
- Widespread use of Incentive Contracting with generous sharing of savings to the government would stimulate contractors to improve
- Increased discipline in holding to stable spiral requirement blocks will avoid requirements creep and speed/improve execution
- Prudent use of Program Stability reserves can keep programs on track while avoiding the need to tap and destabilize other programs when problems arise

What Can We Do About All of This?

- DepSecDef, DCJS, USD(AT&L) & ASD(NII) work out a flexible, iterative, capability based, CAIV-driven revision to the requirements process closely matched to a complimentary evolutionary acquisition process with appropriate feedback opportunities and paths
- USD(AT&L), ASD(NII), & Service SAEs place added emphasis on mod & sim as deliverables, CAIV trades, demos, improved costing methodology, incentive contracts, and more attention to program execution; enforce employment of MOSA
- As the technology matures, increasingly employ MDA in source selection; incentivise its use in program execution and support
- Re-initiate and further refine the Program Stability Fund Process
- Evaluate DSB recommendations on Joint Force Capabilities and expedite incorporation as appropriate

What Can We Do About All of This?
(Continued)

- Develop and expand a DoD System-of-Systems Engineering capability (the Single Integrated Air Picture JSSEO is a good pathfinder example)
- Accumulate ongoing metrics on program overruns, schedule slips; actively keep score, levy goals on Service SAEs and PEOs, etc., periodically review; demonstrate reduced tolerance for overruns