

Defense Acquisition Performance Assessment— The Life-Cycle Perspective of Selected Recommendations

16 June 2008

Dr. Peter Hantos
Software Acquisition and Process Department
Software Engineering Subdivision

Prepared for:

Space and Missile Systems Center
Air Force Space Command
483 N. Aviation Blvd.
El Segundo, CA 90245-2808

Contract No. FA8802-04-C-0001

Authorized by: Engineering and Technology Group

20080807 234

PUBLIC RELEASE IS AUTHORIZED; DISTRIBUTION UNLIMITED

Defense Acquisition Performance Assessment— The Life-Cycle Perspective of Selected Recommendations

16 June 2008

Dr. Peter Hantos
Software Acquisition and Process Department
Software Engineering Subdivision

Prepared for:

Space and Missile Systems Center
Air Force Space Command
483 N. Aviation Blvd.
El Segundo, CA 90245-2808

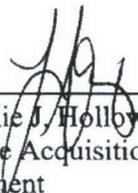
Contract No. FA8802-04-C-0001

Authorized by: Engineering and Technology Group

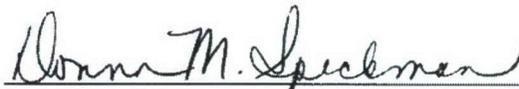
PUBLIC RELEASE IS AUTHORIZED; DISTRIBUTION UNLIMITED

Defense Acquisition Performance Assessment— The Life-Cycle Perspective of Selected Recommendations

Approved by:



Dr. Leslie J. Holloway, Director
Software Acquisition and Process
Department
Software Engineering Subdivision
Computers and Software Division
Engineering and Technology Group



Donna M. Speckman, Director
Research and Program Development
Office
Engineering and Technology Group



Defense Acquisition Performance Assessment – The Life-Cycle Perspective of Selected Recommendations

Dr. Peter Hantos
The Aerospace Corporation

As a significant milestone in the Department of Defense's (DoD) continuous self-assessment process, an important document, the Defense Acquisition Performance Assessment (DAPA) report, was released in early 2006. The report – in its sweeping and integrated assessment – attempted to consider all critical aspects of defense acquisition and made recommendations for each of the major elements of the Defense Acquisition System (DAS). The author's goal in this article is to analyze the conceptual integrity of selected recommendations, using an approach that has been refined during the author's life cycle modeling research. Here, conceptual integrity refers to potential contradictions between the recommended actions that, when viewed independently from each other, appear to be viable. Why the life-cycle modeling focus? Life-cycle models represent the backbone of both acquisition and development processes, and this focus facilitates the analysis of concerns that crosscut in the impacted domains.

On June 7, 2005, Gordon England, Acting Deputy Secretary of Defense, authorized an assessment of the DAS, and created a panel to carry out the DAPA project. A detailed review that covers all aspects of the final report is beyond the scope of this article. Interested readers are invited to study the full text, which can be downloaded from the panel's Web site [1].

Of the panel's recommendations, the following four were selected for discussion on the basis of their life-cycle modeling aspects:

- Allowing program managers to defer non-Key Performance Parameter (KPP) requirements.
- Realigning Milestone B to occur at Preliminary Design Review (PDR) in the Defense Acquisition Management Framework.
- Improving the measurement of technology readiness.
- Making time (schedule) a KPP for the acquisition.

While each of these recommendations appears sound in the abstract sense, their implementation would pose serious challenges. The objective of this article is to identify inherent, life-cycle, structure-related problems with the Defense Acquisition Management Framework that would have to be resolved before attempting to implement the reviewed recommendations.

Because the article is concerned with cross-cutting issues, it did not seem effective to use the traditional approach of reviewing each recommendation in the order in which it is discussed in the DAPA report. Instead, a kind of *reverse* approach has been chosen. A comprehensive, albeit hypothetical, case study of a military space system is presented, and the potential impact of relevant DAPA recommendations on this sample acquisition is explored. The expectation is that the case study will demonstrate implementation

ambiguities intrinsic to the panel's recommendations.

The Current Acquisition System

Figure 1 sets the context of the discussion. The diagram shows the interfaces and interactions among the three processes of the DAS: Planning, Programming, Budgeting, and Execution (PPB&E), Joint Capabilities Integration and Development System (JCIDS), and the *little a* acquisition process outlined in the DoD 5000.2 instruction. The shading in Figure 1 means to further emphasize that the article's analysis is only focusing on panel recommendations that are related to the *little a* dimension of the DAS. Since the case study is a military space acquisition example, a mapping of the DoD 5000.2 Defense Acquisition Management Framework [2] into the National Security Space Acquisition Policy 03-01 (NSSAP) acquisition phases [3] is needed. This mapping is shown in Figure 2 (see page 26). Note that the major phase gates are called *milestones* in DoD 5000.2 but are referred to as Key Decision Points (KDPs) in NSSAP 03-01. The content of the technical reviews is the same, as their names are similar, and all represent system-level reviews. In DoD 5000.2, these reviews are as follows: System Requirements Review (SRR), System Functional Review (SFR), PDR, and Critical Design Review (CDR). In NSSAP 03-01, System Design Review (SDR) replaces SFR. In both processes, IOC represents Initial Operational Capability.

NSSAP 03-01, unlike DoD 5000.2, distinguishes between two acquisition models. One, the Small Quantity Model, is slated for the acquisition of the majority of space assets. The second, the Large Quantity Production Focused Model, is used for the acquisition of user equipment, terminals, etc. In Figure 2, the mapping for the Small Quantity Model is presented. What makes

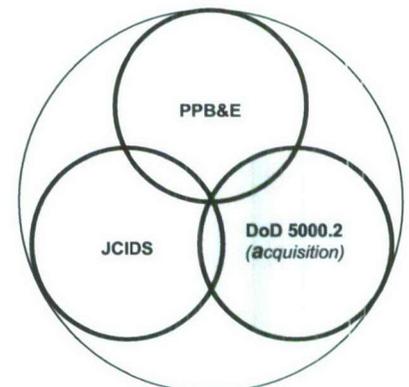
space systems different from the majority of weapon systems? First, they are highly software-intensive. Typical ground control systems have millions of lines of code, and even the spacecraft and satellite payload segments could easily contain a half-million lines of code. Second, satellite systems, along with their ground stations and boosters, are usually acquired in quantities of 10 or less due to the high expense of satellites and launch costs. These systems are practically custom-built rather than *mass-manufactured*, hence the need for the Small Quantity Model.

Space System Acquisition Case Study

This system would ultimately replace an existing network of military satellites that is slowly becoming obsolete. New, critical capabilities are planned. The final system in space would manage mixed missions, generations, and constellations of satellites. On the ground, a complex network of space/ground connections, mobile and permanent ground stations, and command and analysis centers are envisioned.

Evolutionary Acquisition (EA) has been chosen as the acquisition strategy. EA is

Figure 1: Interaction Among PPB&E, JCIDS, and DoD 5000.2



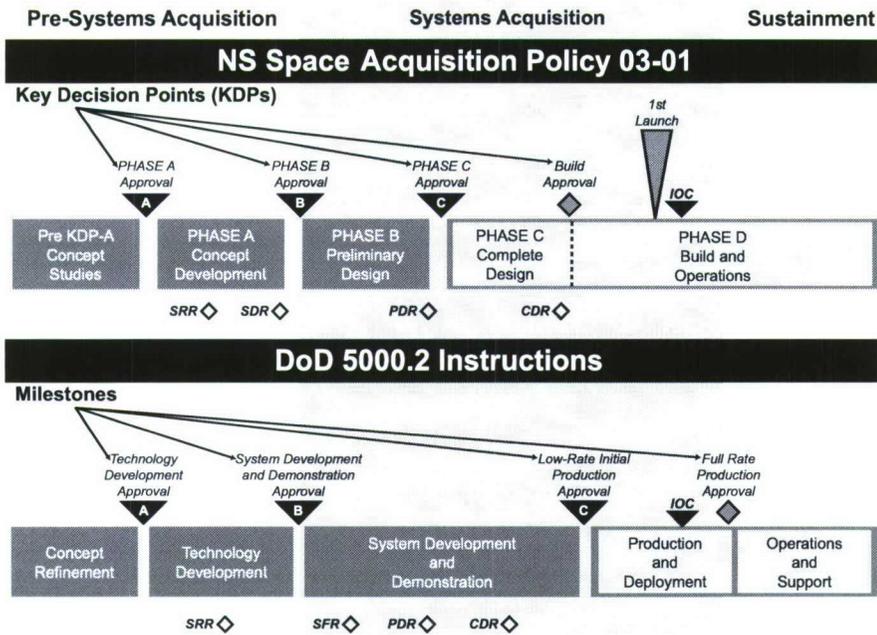


Figure 2: Mapping of DoD 5000.2 Into NSSAP 03-01

defined as an acquisition approach that delivers capability in an incremental fashion, recognizing the up-front need for future capability improvements. These future capabilities are to be contracted and delivered in the context of successive acquisition increments (Figure 3). As part of the acquisition strategy it is also decided that the contract in the first increment of the acquisition (to be referred as First Acquisition Increment) would have two major deliveries, in effect calling for the development and delivery of two system increments'. The planned content of these System Increments is as follows: The entire ground system (except for future mobile stations) would be developed in the first system increment. The operational acceptance test of this new ground system would involve the full control of selected, existing constellations. All new space assets (spacecraft and payload hardware/software) and the mobile stations would be delivered in the second system increment.

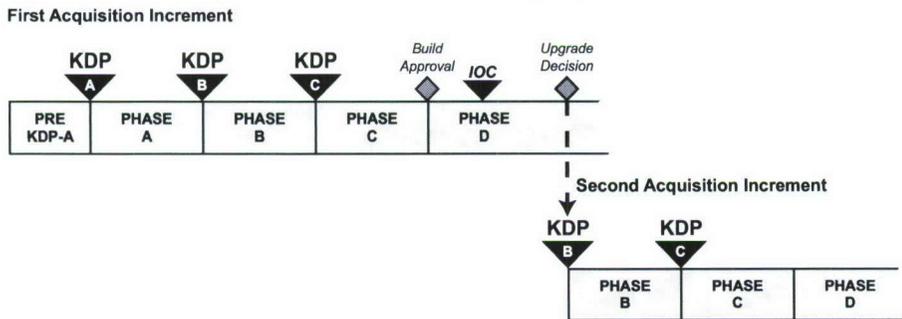
The plan is to first launch only a few prototypes of the new satellites, then decide

about the acquisition of more satellites later. New requirements are expected for the ground system on the basis of experience gained during the launch and operation of the prototype satellites. Most likely, other mission and satellite payload capability requirements will also emerge, triggering the need for a generation of new satellites.

The program's acquisition strategy outlines a plan for soliciting bids from up to three contractors during the Pre-KDP-A Concept Study Phase, down-selecting to two at KDP-A, and making a final decision at KDP-B. This is an expensive but highly risk-averse strategy to mitigate contractor uncertainties. Figure 4 illustrates a simplified life-cycle model, accommodating the first acquisition increment.

Figure 4 depicts several concurrent streams of events and their relationships, showing a notional alignment of the Milestone Decision Authority (MDA) actions with the decision-obligation-spending sequences of the PPB&E process. Congress allocates money for only one year's worth of activity. So PPB&E is repeated

Figure 3: Successive Acquisition Increments in Evolutionary Acquisition



every year, and the appropriated funds, even though they belong to the same program, are in different spending states depending on when they were approved.

An explanation of the depicted contract actions is as follows: The program would require a Lead System Integrator (LSI) – sometimes called the *Prime* if the main contractor performs development tasks as well – and the contributions of, most likely, several sub-contractors. During the Pre-KDP-A period, three contractors are to provide concept studies. Following an evaluation of these studies, the MDA invites only Lead-2 and Lead-3 to continue. In Phase A only the potential leads compete, but upon entry to Phase B the selected lead chooses sub-contractor partners, hence the change to *team* designation.

With respect to funding, a naïve assumption is that work would only start after the budget and contracts are secured. In reality, companies that want to stay in the game have to be involved in continuous research and technology development even before the solicitations go out, and the funding of such activities must come from internal resources. These technology development and miscellaneous research activities are not shown in detail. For example, to bid for this project, Lead-1 (who ultimately was not invited to continue in Phase A) would already be engaged in relevant development activities. The same is true for potential sub-contractors. In Figure 4, the blocks with upward diagonal shading represent this early engagement. Some of the efforts during bidding are covered by the government, but it is not unusual for companies to pay for their expenses in an expectation of winning a lucrative long-term contract.

Study of the technical reviews in the overall life-cycle structure results in further controversies. These reviews – holdovers from the long-defunct Military Standard (MIL-STD)-1521B – are based on the Waterfall process, because in 1985, at the time of the last update of the standard, Waterfall was the only approved development life-cycle model for the DoD. (For further details, see [4].) For example, SDR is supposed to be a technical review of the system design supporting the MDA's decision-making at KDP-B, the entry to the preliminary design phase. The fact that system review is supposed to precede the start of preliminary system design is confusing, and neither the phase nor the review name/content is consistent with reality. Planning and conducting system PDR in Phase B is problematic as well. In Phase B, design and development of all segments progresses at different paces; total, vertical synchronization of reviews (i.e., lining up segment-level design

reviews for ground software, spacecraft software, spacecraft hardware, payload software, etc.) is simply not feasible. The first ground system increment must be almost ready for integration, and there must be substantial progress on the spacecraft and payload side as well. By the time system CDR comes, the disconnect is even more striking. The life-cycle modeling-based analysis shows the root cause for this disconnect. The first increment of the acquisition is a sequential structure by design, which via its naming conventions and phase descriptions enforces a Waterfall development life cycle. Such a life-cycle model is clearly inappropriate for a large scale, concurrent engineering project.

Figure 4 shows Spiral as the life-cycle model of choice for ground software development. Both DoD 5000.2 and NSSAP 03-01 state that Spiral Development (SD) is one of the main processes that perform EA. Are the depicted ground spirals what the government policies refer to? The answer is an unqualified *no*. From the earliest days, the prevailing misconception is that DoD 5000.2 is spiral development, where concept refinement is the first spiral, technology development is the second one, system development and demonstration is the third one, and so on. Also, entry criteria for every milestone (or for the corresponding KDPs in NSSAP 03-01) include required risk management activities (risk identification, risk reduction, and risk mitigation plans), reinforcing the notion that we are performing SD. At the same time, looking at the concise definition of the Spiral and its essential characteristics [5], it becomes clear that these activities are not what the successful application of spiral concepts assumes. The key risk-related mechanism that is unique to SD is embodied in (a) the concurrent engineering of all artifacts and (b) the risk-driven planning of the content, and consequently cost and schedule successive spirals. Having risk mitigation plans in the conventional sense is different from spiral planning. It involves the creation of additional plans to eliminate or gradually reduce the risk by having alternative course(s) of actions lined up in case the risk materializes or its likelihood drastically increases. A key element of such risk planning is that funding for alternative actions needs to be provided in addition to the allocated, regular cost of development.

The applied SD method in this case study is a highly localized and not a system-level process, and it is not supportive of this program's EA strategy. While a detailed discussion of EA is beyond the scope of this article, some justification for this statement is needed. As NSSAP 03-01 states, during SD that supports EA, a desired capability is identified, but the end-state

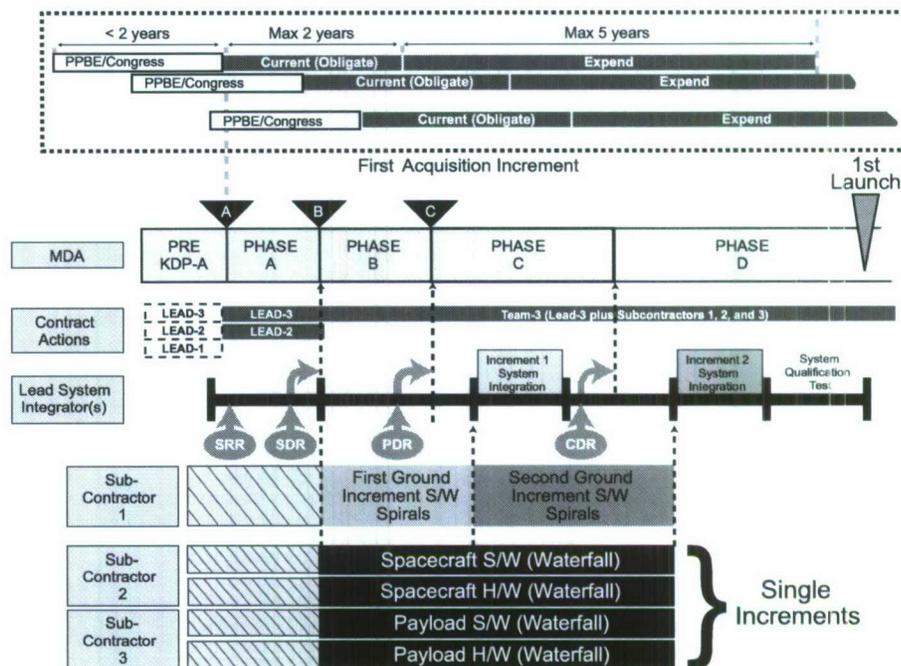


Figure 4: Simplified Life-Cycle Model

requirements are not known at program initiation. In our case study, not only system capabilities but detailed system requirements are also known prior KDP-B. In fact, even the high-level requirements for the two software increments are determined in advance and go on contract as well. Also, looking at Figure 3, it is becoming clear that development spirals (iterations) carried out during Phase B or even Phase C are far removed from the upgrade decision that triggers the second acquisition increment. The upgrade decision – besides new, emerging requirements – should be based on the status of current technology and user experiences gained during the operational phase and not on information gathered during earlier development spirals. The reader might also wonder, if this is the case, why SD was chosen by the case study's program manager for ground software development. Was it an arbitrary decision and was it a mistake? On the contrary, iterative development is the prudent strategy for this kind of large scale, concurrent engineering project, and SD is a well-known, *brand-name* iterative method. Quoting Martin Fowler's whimsical advice, *You should use iterative development only on projects that you want to succeed ...* [6].

As pointed out earlier, the acquisition life-cycle phases, the management commitment points, and their associated mandatory documentation represent a Waterfall sequence from the point of view of system development. This inability to reconcile the conflicting acquisition and development life-cycle models is one of the main reasons for the poor track record of the Spiral Model in

defense acquisitions. In summary, applying spiral development in an acquisition increment to manage risks could be an effective project management strategy, but this strategy has nothing to do with the spiral process assumed in DoD 5000.2 or, for that matter, in the Defense Authorization Act of fiscal year 2003 that further specifies mandated characteristics of spiral development for major defense acquisition programs [7].

Deferral of Non-KPP Requirements

Allowing program managers to defer non-KPP requirements to later upgrades is an attractive proposition from the program manager's view. It provides an effective risk management tool by greatly expanding their decision-making authority and flexibility. In the context of our case study, how could the program manager using this newly acquired freedom reduce the scope of the first acquisition increment? Unfortunately, analysis shows there are not many opportunities after all. One possibility is to make the delivery of the first spiral of the ground system the first acquisition increment. This is a useful and complete capability (controlling the existing constellation of satellites), but it does not provide enough value to the customer, since there was already an operational system in place. In other words, the delivery of this new but compatible ground system is an excellent engineering objective, but insufficient as an acquisition objective. Also, it is not clear what we would do about spacecraft and payload development. They can not be deferred until after the delivery of the first increment of the ground system; that would

push out the availability of new satellites with new capabilities to an unacceptable, distant period. On the other hand, if their development is started simultaneously with the ground system, at the time of ground system delivery they would be still in an incomplete, intermediate state of their Waterfall process-streams. Receiving documentation, prototype breadboards and models, and maybe some untested code would not be an acceptable acquisition value proposition either.

There are other considerations that would make the deferral of requirements difficult. For example, complex graphics and elaborate display designs are important in any ground system. As a requirements-pacing strategy, one might consider releasing the first version of the ground software with simplified user interfaces. This is an effective engineering approach, but it may backfire with end-users of the system. In similar situations, satellite operators forced to work with intermediate systems having limited capabilities created resentment and blocked buy-in when the final system became available.

In conclusion, the opportunity for delaying non-KPP requirements is great, but complex space systems might not always lend themselves to a feasible granularity of requirements for such deferral.

Realignment of Milestone B

This DAPA recommendation calls for the realignment of Milestone B to occur at PDR, and the justification is as follows: The greatest trade space and the largest risk reduction opportunities exist between

Milestone A and Milestone B, and the DoD places most program focus on Milestone B, because premature technology and system design decisions at Milestone B lead to technical problems during system design and development. Unfortunately, the term *realignment* is ambiguous due to lack of implementation details. Using the equivalent NSSAP 03-01 terminology, it needs to be clarified whether KDP-B should be moved forward or PDR moved backward (Figure 5).

Again, the phase definitions and reviews are in conflict. The declared objective of KDP-B is to gauge entry into Phase B. This phase-gate objective would indicate that we cannot talk about the move of KDP-B, only the move of PDR. However, if Phase C's objective is complete design, then PDR must immediately precede it. Moving up PDR means that its successful completion would lead us to complete design activities during a phase that is only designated for preliminary design. Finally, we are left with the delicate but unanswered question of positioning CDR. Would CDR move up as well? The unfortunate conclusion – again – is that the root cause of the problem is the ingrained Waterfall that is imposed on the developer by the acquisition models, and the planned move of decision points or reviews would not help either the MDA or the program manager.

Technology Readiness

As mentioned earlier, technology was identified as an important focus area for the DAPA inquiry. The findings state that

there are no clearly definable measures of technology readiness, and the inability to define and measure technology readiness during Technology Readiness Assessments (TRAs) is the reason that immature technology is incorporated into plans prior to Milestone B. On the contrary, numerous sources are available to help with technology readiness assessments (see, for example [8], [9], and [10]). These referenced materials provide a workable version of Technology Readiness Levels (TRLs), applicable to the hardware elements of Ground, User, and Launch Segments of space systems. Even though there is some ambiguity regarding the use of these TRLs for assessing software in general and the hardware elements of the Space Segment in particular, still, measuring technology readiness should not be the main concern. While the exploration of all issues is beyond the scope of this article, the examination of the life-cycle dimension of TRA highlights the following, inherent problem of the Defense Acquisition Management Framework.

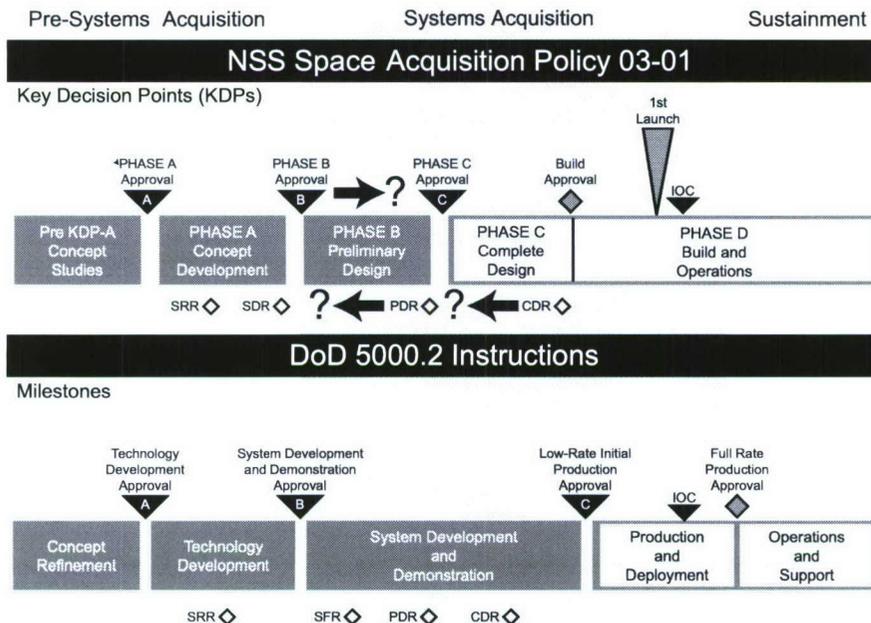
The applicable DoD policy for technology maturation at Milestone B is unambiguous (Chapter 5.3 of the DoD desk-book on TRA [12]): *All Critical Technology Elements (CTEs) should be identified and successfully demonstrated on a TRL 6 or higher before Milestone B.*

The concern relates to the execution of this policy. This simplified case study shows five concurrent engineering streams: ground software, spacecraft hardware, spacecraft software, payload hardware, and payload software (user systems and launch systems are also important segments of a total space system solution but were omitted for simplicity's sake). A TRA must be conducted for all segments in all domains. It is fair to assume that if KDP-B is the one and only phase gate to exit from concept development, then the enabling, critical technology elements of all concurrent processes must be at high TRL. Is this a reasonable assumption? What happens if some of the technologies are riskier than others and do not mature at the same pace? Clearly, this imbalance of concurrent engineering streams puts the predictability of the overall program in jeopardy. Or, theoretically, design of critical parts for the whole program could be forced to idle until the resolution of delinquent technology issues in the affected segments is completed, but that is obviously not a feasible option either.

Time Certain Development

One of the recommendations would

Figure 5: Realignment Possibilities for KDP B



declare Time Certain Development as the preferred acquisition strategy by making time a KPP for the acquisition. First, when programs at Milestone A would be required to be budgeted on the basis of high-confidence estimates. Second, when the time-to-need and the current technology risk level are determined, the program should be time-constrained. Finally, technical performance should be traded-off to maintain this schedule. (see page 51 of the DAPA Report [1]). However, cost and schedule estimation in the presence of technology risks is difficult for various reasons. Theoretically, in all conventional parametric cost estimation models, cost, schedule, and performance can be seamlessly traded (although, this trade only works for routine, repeatable activities – in the case of software, for coding). The models establish an exponential relationship between performance and cost, and also between cost and schedule, to facilitate this trade. Fred Brooks pointed out an important and frequently overlooked fact in his classic book [11] that when a task cannot be partitioned because of sequential constraints, the application of more effort has no effect on the schedule. In terms of technology development, the process is inherently a sequence of learning steps, building on the results of previous experiments. This sequential process of experimentation and learning, combined with the probabilistic nature of success, make the implementation of Time Certain Development very problematic.

Conclusion

The acquisition life-cycle models of the DoD/NSSAP policies are inherently Waterfall, and as such, inadequate for the acquisition of large-scale, software-intensive systems, even if they are used with the intent of EA. The concerns raised in the case study indicate that consideration for an additional DAPA focus area, engineering, would be required to develop feasible changes to the DoD 5000.2 and NSSAP 03-01 policies. One can speculate that the absence of engineering considerations in the recommendations for industry is intentional; reflecting a hands-off approach by not constraining the contractor's engineering solutions. It is indeed desirable not to proscribe engineering processes in acquisition policy documents. Nevertheless, the case study convincingly demonstrates that current – not even state-of-the-art, but certainly state-of-the-practice – engineering methods, particularly integrated life-cycle models of concurrent engineering and iterative development, represent severe, hidden con-

straints, and they would have to be considered as key influencing factors during reworking the *little a* acquisition system.

Finally, the panel recommends the use of system dynamics to analyze the internal relationships of the acquisition system [12]. System dynamics is a modeling approach to studying complex systems via the identification and simulation of internal feedback loops of the system [13]. System dynamics is indeed the right tool for analyzing the tension resulting from unintended consequences of conflicting behaviors, but one could argue that before such a sophisticated and complex tool is unleashed, analyzing the life-cycle model structure of development should be satisfactory for identifying some fundamental, systemic conflicts. ♦

Acknowledgements

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

- **Reviewers:** Suellen Eslinger, Mary Jo Gura, and Dr. Leslie Holloway, The Aerospace Corporation.
- **Sponsor:** Michael Zambrana, U.S. Air Force Space and Missile Center.
- **Funding Source:** Mission-Oriented Investigation and Experimentation Research Program, Software Acquisition Task.

References

1. DoD. DAPA Project. Mar. 2006 <www.acq.osd.mil/dapaproject/>.
2. DoD. "DoD 5000.2 Instruction on the Operation of the DAS." May 2003 <www.akss.dau.mil/darc/darc.html>.
3. United States Secretary of the Air Force. "National Security Space Acquisition Policy 03-01." Dec. 2004.
4. Hantos, P., "System and Software Reviews Since Acquisition Reform." Southern California Software Process Improvement Network, Apr. 2, 2004 <www.ucsc.edu/spin/media/pdf/SoCal-SPIN_PH_slides-04-02-04.pdf>.
5. Boehm, B., and W.J. Hansen. "The Spiral Model as a Tool for Evolutionary Acquisition." CROSSTALK May 2001 <www.stsc.hill.af.mil/crosstalk/2001/05>.
6. Fowler, M. UML Distilled. 2nd ed. Addison-Wesley, 2000.
7. United States House of Representatives. FY2003 Defense Authorization Act. Section 803 of 116 STAT. 2604 Public Law 107-314.
8. Mankins, J.M. "Technology Readiness Levels – A White Paper." National Aeronautics and Space Administration, 1995 <www.hq.nasa.gov/office/codeq/trl/trl.pdf>.

9. Graettinger, C.P., et al. Using the Technology Readiness Levels Scale to Support Technology Management in the DoD's ATD/STO Environments. CMU/SEI-2002-SR-027. Sept. 2002.
10. "Department of Defense Technology Readiness Assessment (TRA) Desk book." May, 2005 <www.akss.dau.mil/darc/darc.html>.
11. Brooks, F.P. The Mythical Man-Month – Essays on Software Engineering. Addison-Wesley, 1982.
12. Venture Services, LLC. "Monitor Government." Appendix D – Defense Acquisition Performance Assessment Project Report, A Baseline Literature Review. DAPA Project, Mar. 2006 <www.acq.osd.mil/dapaproject/>.
13. Sterman, J.D., Business Dynamics: System Thinking and Modeling for a Complex World. McGraw-Hill, 2000.

Note

1. Here, increment is used in two different contexts as is common in current software standards. In acquisition increment refers to contractual and user concerns, while in development increment refers to engineering and implementation concerns.

About the Author



Peter Hantos, Ph.D., is a senior engineering specialist at The Aerospace Corporation. He is the principal investigator of the Unified Life Cycle

Modeling research, an effort to introduce comprehensive modeling and simulation approaches to software-intensive system development. Hantos has more than 30 years of experience as a professor, researcher, software engineer, and manager. Prior to joining Aerospace, as principal scientist at the Xerox Corporate Engineering Center, he developed corporate-wide engineering processes for software-intensive systems. Hantos holds masters of science and doctorate degrees in electrical engineering from the Budapest Institute of Technology, Hungary.

The Aerospace Corporation
P.O. Box 92957 - MI/112
El Segundo, CA 90009-2957
Phone: (310) 336-1802
Fax: (310) 563-1160
E-mail: peter.hantos@aero.org