Mission Assurance Through Improving Software Technology Readiness Assessments (TRA)s for National Security Space (NSS) Systems

Daniel Winton
Sr. Engineering Specialist
The Aerospace Corporation
Mission Assurance Subdivision

Software Development is Broken

“The acquisition of large, complex software-intensive systems has historically been fraught with major problems, including performance deficiencies, extensive software defects, and cost and schedule overruns.”

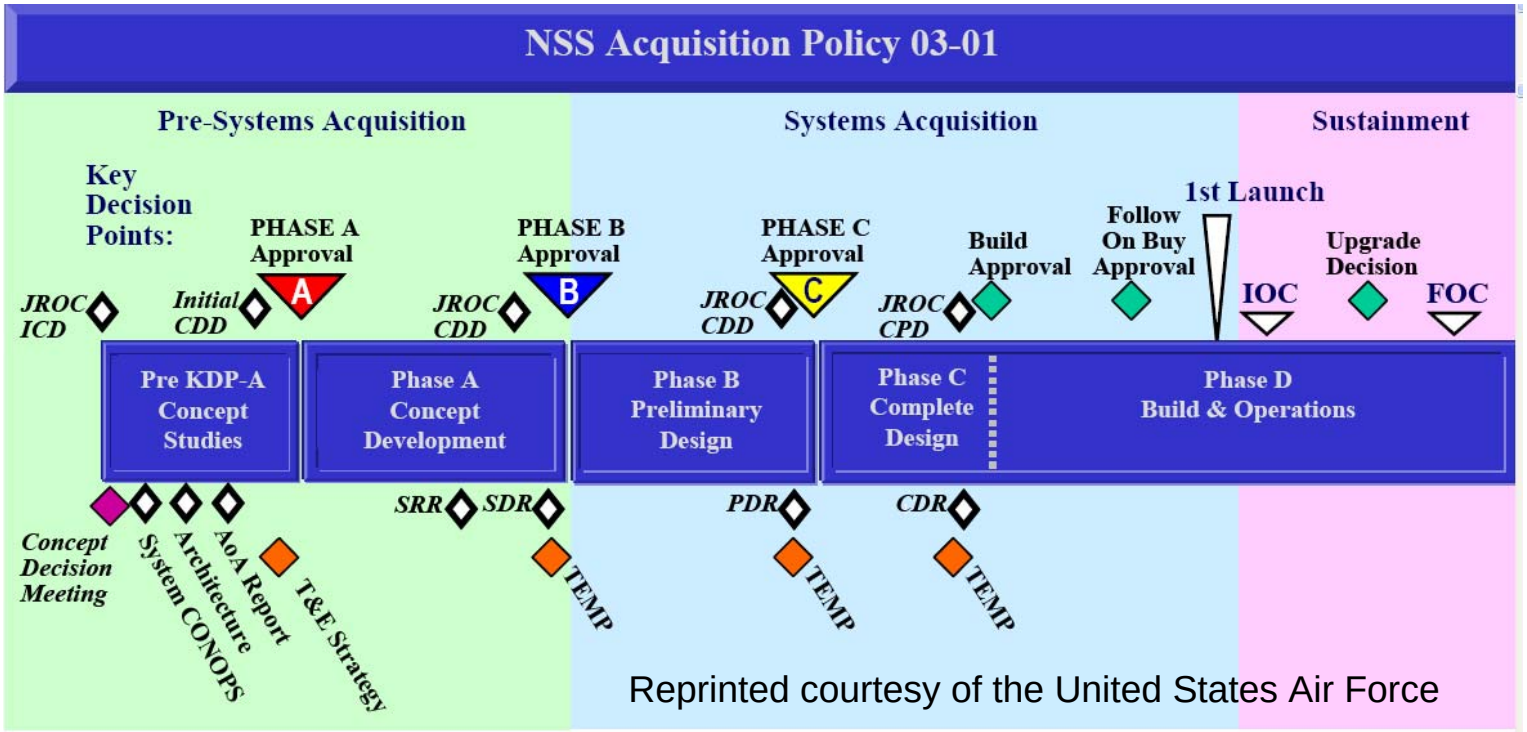
Richard Adams
Aerospace TR-2006(8550)-1

Thesis

Improving how systems software is addressed in Technology Readiness Assessments (TRA)s improves the likelihood of success of National Security Space (NSS) System acquisitions

Acquisition Review: What is NSS 03-01?

- National Security Space Acquisition Policy 03-01 is the space acquisition policy guidance for the DoD
- The Technology Readiness Assessment (TRA) process must fit into this guidance



Reprinted courtesy of the United States Air Force

Acquisition Review: What is a TRA?

- **A TRA is a snapshot of the maturity of the technological components that are critical to the success of a system**
  - Critical Technology Elements (CTE)
- **A CTE's readiness is assessed by using a set of Technology Readiness Levels (TRL) that reflect its level of maturity.**
- **TRAs are done at key decision point (KDP) milestone decisions**
  - A critical technology's maturity affects the decision as to whether the acquisition can move from one phase to another

Current Software TRL Descriptions

- **TRL-1 Description**
  - Lowest level of software technology readiness. A new software **domain** is being investigated by the basic research community. This level extends to the development of basic use, basic properties of software architecture, mathematical formulations, and general **algorithms**.
- **TRL-2 Description**
  - Once basic principles are observed, **practical applications can be invented**. Applications are speculative, and there may be no proof or detailed analysis to support the assumptions. Examples are limited to analytic studies using synthetic data

What is Wrong with the Software Technology Readiness Assessments

- **Most domains and algorithms are not considered new**
  - New software domains, e.g. Router in space
    - Router done terrestrially, therefore, not really new
  - New algorithms , e.g. New Kalman filter
    - Variation on existing algorithm, therefore, not really new
- **Software readiness becomes relevant only in context of the system to which it will be applied**
  - The following statement is irrelevant

    Once basic principles are observed, practical applications can be invented

- **TRLs 1 and 2 are foundational to determining if software is a critical technology element**
  - But they often don't apply

Options

- **Eliminate software from Technology Readiness Assessments**
  - We still have
    - Cost and schedule overruns
    - Immature software
    - Software to hardware integration problems
    - This is not a good choice
- **Redefine what TRAs mean by new software technology in acquisitions**
  - We are not looking at new domains or algorithms
  - We are not observing basic principles or looking to invent practical applications
- **Something else must be new**

Problem Expansion Heuristic

- ***Sometimes, but not always, the way to solve a difficult problem is to expand it.***
  (Rechtin, 1991, 45)

So, what are we building?

- **We're building NEW Software Intensive SYSTEMS**
- **With each development, two aspects of software are, by definition, new**
  1. The system as a whole
  2. The interfaces
     - If for no other reason, their complexity makes them new
     - No single component individually has to be considered new

Systems Software as a New Technology

- **If we assume the systems software as a whole and the external and internal interfaces are new**
  - These items should always be default systems software CTEs
  - Additionally, all individual software components should be candidate CTEs unless proven otherwise
- **Even if there are similarities with a previous effort, there are still unique aspects within each systems software development**

Space System Software Development Effort Unique Aspects

- A newly formed software development team will be working on the system
- The type of software being developed has not been developed by the software development team before
- A software language which the team has not used is being proposed
- A software development paradigm with which the team is not familiar such as object oriented (OO) is being used
- New operating system is being utilized
- Computational hardware that is different than that which was used before
- New system hardware such as payload and bus components
- Interfaces that are new or different
- A new or unproven software architecture
- Use of commercial off-the-shelf (COTS) that may have changed or may be being utilized in a different way

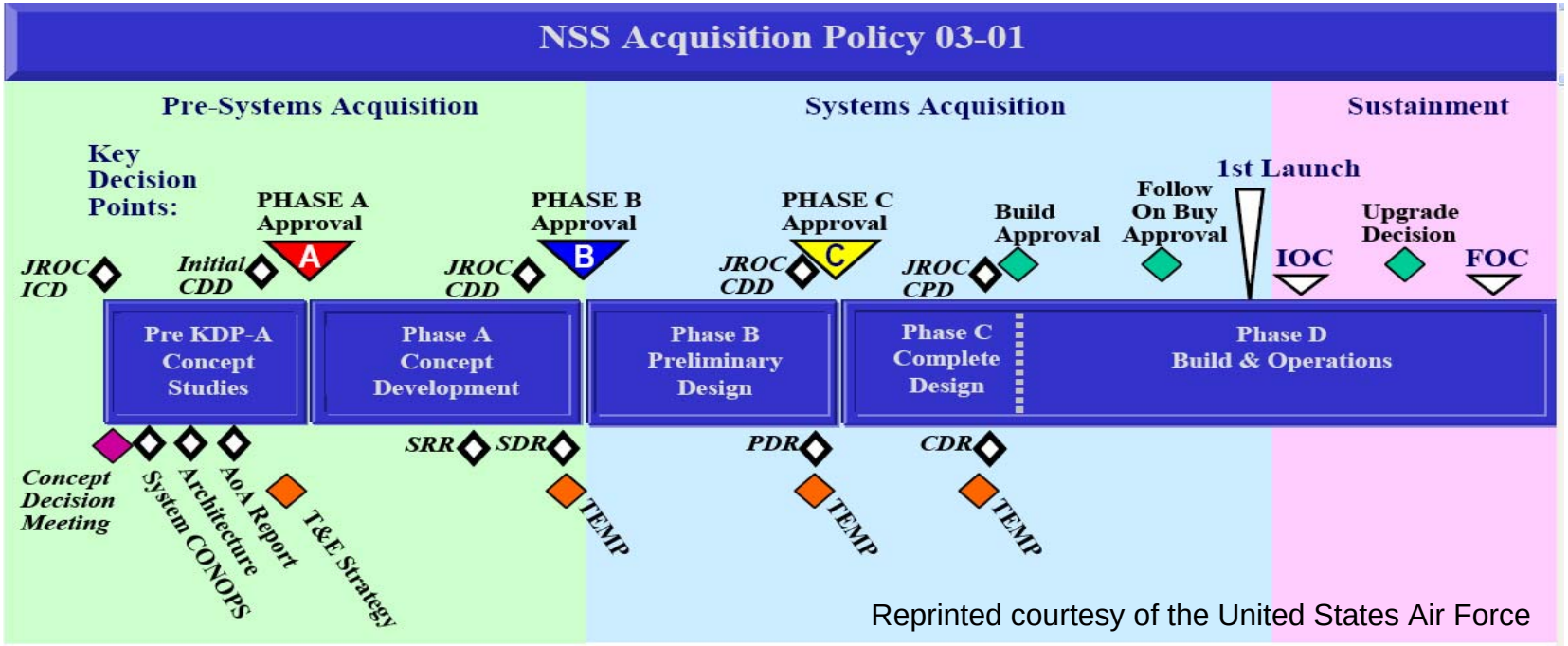
Recommended New TRA Definitions

- **Software technology readiness is different for systems than for domains and algorithms**
  - New definitions are needed
- **Systems Software (SS)**
  - Critical Technology Element (**SS-CTE**)
  - Technology Readiness Level (**SS-TRL**)
- **Domain Software (DS)**
  - Software definitions currently identified in TRA for Domains and Algorithms
    - Critical Technology Element (**DS-CTE**)
    - Technology Readiness Level (**DS-TRL**)

Phases Where SS-TRL Definitions Will be Applied and What They Do

- **Phase 0 (Pre KDP A): Concept Studies**
  - SS-TRLs 1 and 2
    - Problem identification
    - Requirements identification
- **Phase A: Concept Development**
  - SS-TRLs 3, 4, 5 and 6
    - Risk mitigation
- **Phases B-D: Design and Development Phases**
  - SS-TRLs 7, 8, and 9
    - Gauge for development progress

NSS 03-01/SS-TRL Map



Reprinted courtesy of the United States Air Force

Proposed SS-TRLs and the system acquisition phases

SS CTEs must be at the SS TRL immediately to the right of a KDP

System Acquisition									
Phase	Phase 0		Phase A				Phase B	Phase C	Phase D
TRL Level	SS-TRL-1	SS-TRL-2	SS-TRL-3	SS-TRL-4	SS-TRL-5	SS-TRL-6	SS-TRL-7	SS-TRL-8	SS-TRL-9

KDP-A KDP-B KDP-C

High Level View of Changes Needed in TRAs

- **SS-CTEs**
  - Always include the system as a whole and the interfaces
  - Assume each software component is a candidate SS-CTE until proven otherwise.
- **Embed software development best practices in SS-TRLs in order to evaluate SS-CTEs**
  - Pre Phase A (Phase 0)
    - Assure that a viable integrated hardware/software architecture is well defined and the problem is well understood
      - Government Documentation (GRA, CONOPS, OCD
  - Phase A
    - Lower risk through use of engineering models (EM)
    - Assure that the solution is well defined
      - Contractor Documentation (SSDD, SDP)
  - Phases B and C
    - Validate software through simulation

High Level View of Embedded Software Development Best Practices in SS-TRLs

- **Create viable integrated hardware/software architecture**
  - From the beginning software involved in hardware and algorithm development
  - Don't try to apply software to hardware
- **Assure government and contractor documents are adequate**
- **Software engineering models**
  - Equivalent to breadboard in hardware
  - Experimentation reduces risk
- **Use simulation to validate solution**
- **Early development of vocabulary for interoperability**
- **Watch COTS, NDI and Reuse**
- **Limit requirements creep**

You are at this SS-TRL When…

Ph	No	Most significant aspects
0	1	▪ System DOTMLPF performed ▪ Government Acquisition team engaged
	2	▪ Government documentation established (GRA, CONOPS, OpCon) ▪ Government chief systems engineer and government chief software engineer agreement that potential system solution is viable
A	3	▪ Contractor documents (SDP, SSDD), and Engineering Models (EM) initiated
	4	▪ Updated government documents and contractor documents and EMs
	5	▪ EM software components integrated and put under configuration mgmt. ▪ EM software ready for integration with hardware
	6	▪ Testing of EM in a high-fidelity laboratory environment or in a simulated operational environment
B	7	▪ Feasibility of integrated hardware/software solution demonstrated
C	8	▪ Integrated hardware/ software solution fully operational, documentation complete
D	9	▪ Actual system proven through successful mission operations

Conclusion

- **Systems software should be assessed in TRAs along with new software domains and algorithms**
  - Doing so ensures that the systems software is considered as a part of KDP decisions
- **Embedding software development best practices in TRA assessments results in**
  - Greater confidence in software development estimates
    - Potential to improve from 50% confidence level to 80% confidence level
  - Less cost and schedule overruns
  - Higher quality software

Backup

Why Apply Software Development Best Practices to SS-TRL Definitions

- **Ensures that the system architecture, system documentation and engineering models are evaluated with respect to SS-CTEs**
- **Evaluating SS-CTEs affects the TRA** and influences the KDP milestone decisions
  - The state of systems software is understood before moving from one acquisition phase to another

SS-TRL-1

- **Definition**
  - New or novel software development effort initiated as a part of an integrated hardware/software solution.
- **Description:**
  - DOTMLPF performed
  - The government acquisition organization has been engaged.
  - Lowest level of systems software technology readiness. This extends to ground, bus and payload solutions
- **Supporting Information**
  - JCIDS responsibility
  - Only the system as a whole and the interfaces need to be identified at this point

SS-TRL-2

- **Definition**
  - Government reference integrated hardware/software architecture initiated
- **Description**
  - Government documentation established (GRA, CONOPS, OpCon)
  - Architecture studies and trade studies have been performed. Common System Terminology Dictionary (CSTD) Established. Architecture and CSTD is acceptable from the perspectives of all stakeholders. (Everyone can be equally dissatisfied)
- **Supporting Information**
  - Acquisition Team involvement
  - Government chief system engineer and chief software engineer agreement

SS-TRL-3

- **Definition**
  - Early engineering models (EM)/ simulations and initial documentation describing the hardware and software
- **Description**
  - More architecture studies and trade studies have been performed. Common System Terminology Dictionary (CSTD) Established. Architecture and CSTD is acceptable from the perspectives of all stakeholders. (Everyone can be equally dissatisfied)
  - Initial System Subsystem Design Description has been established by contractor(s) along with a Software Development Plan as a part of the integrated hardware/software architecture.
  - System architecture diagram augmented around critical technology element with critical performance requirements.
  - Simulation/ Stimulation (Sim/Stim) Laboratory buildup and validation plan (SSLBVP) has been initiated.
  - Early prototypes have been created
- **Supporting Information**
  - government chief software and systems engineers say SSDD and SDP are robust enough to allow for successful software development

SS-TRL-4 Description

- **Definition**
  - Updated documentation and EM/ simulations
- **Description**
  - Updated System Subsystem Design Description and Software Development Plan have been generated. SSLVBP is complete. A specific management plan for handling all Non-Developmental Item (NDI) software and its version upgrades and governance issues has been provided.
  - Components of Increment 1 of software baseline have been established and validated.
  - Simulation/ Stimulation (Sim/Stim) Laboratory buildup and validation plan (SSLBVP) has been written.
  - Simulation environment has been established and validated. Early Prototype of components to demonstrate different aspects of eventual system have been successfully run. Emulation with current/ legacy elements has been performed as appropriate.
  - A documented approach to transitioning from prototype to actual system development will has been established and validated
  - COTS/GOTS in the system software architecture are identified and compliance tested.

SS-TRL-4 Supporting Information

- **Definition**
  - Updated documentation and EM/ simulations
- **Supporting Information**
  - Software run in a simulated environment, instrumented components operating in showing validation of critical functionalities.
  - Advanced software development, stand-alone prototype performing under stressing conditions or standalone prototype processing fully representative data sets

SS-TRL-5

- **Definition**
  - Module/ subsystem EM/ simulations in a simulated relevant environment
- **Description**
  - Point at which software technology is ready to start integration with existing systems. The prototype implementations conform to target environment/ interfaces. Experiments with realistic problems. Simulated interfaces to existing systems.
  - Basic increment 1 software components are integrated to establish that they will work together.
  - They are relatively primitive with regard to efficiency and robustness compared with the eventual system.
  - Updated integrated simulations prepared and performed.
- **Supporting Information**
  - Software placed under configuration management.

SS-TRL-6

- **Definition**
  - System/ subsystem model or EM/ simulation demonstration in a simulated relevant environment
- **Description**
  - Representative model or prototype system, which is well beyond that of TRL-5, is tested in a simulated relevant environment. Represents a major step up in a technology's demonstrated readiness. Examples include testing a prototype in a high-fidelity laboratory environment or in a simulated operational environment.
- **Supporting Information**
  - Results from laboratory testing of a prototype package that is near the desired configuration in terms of performance, including physical, logical, data, and security interfaces. Comparisons between tested environment and operational environment analytically understood. Analysis and test measurements quantifying contribution to system-wide requirements such as throughput, scalability, and reliability. Analysis of human-computer (user environment) begun.

SS-TRL-7

- **Definition**
  - System prototype/ simulation demonstration in a high fidelity simulated operational environment
- **Description:**
  - Point at which the program feasibility of a software technology is demonstrated. This extends to operational environment prototype implementations where critical technical risk functionality is available for demonstration and a test in which the hardware/software technology is well integrated.
- **Supporting Information**
  - Critical technological properties are measured against requirements in a simulated operational environment.

SS-TRL-8

- **Definition**
  - : Actual system completed and qualified through test and demonstration
- **Description**
  - Point at which integrated hardware/ software technology is fully operational. Software development documentation is complete. All functionality tested in simulated and operational scenarios.
- **Supporting Information**
  - Published documentation and product technology refresh build schedule. Software resource reserve measured and tracked.

SS-TRL-9

- **Definition**
  - Actual system proven through successful mission operations
- **Description**
  - Actual application of the technology in its final form and under mission conditions, such as those encountered in operational test and evaluation (OT&E). Examples include using the system under operational mission conditions. All software documentation verified. Successful operational experience. Sustaining software engineering support is in place.
- **Supporting Information**
  - OT&E reports.

DS-TRL-1

- **Definition**
  - Basic principles observed and reported
- **Description**
  - Lowest level of software technology readiness. A new software domain is being investigated by the basic research community. This level extends to the development of basic use, basic properties of software architecture, mathematical formulations, and general algorithms.
- **Supporting Information**
  - Basic research activities, research articles, peer-reviewed white papers, point papers, early lab model of basic concept may be useful for substantiating the TRL level.

DS-TRL-2

- **Definition**
  - Technology concept and/or application formulated
- **Description**
  - Once basic principles are observed, practical applications can be invented. Applications are speculative, and there may be no proof or detailed analysis to support the assumptions. Examples are limited to analytic studies using synthetic data
- **Supporting Information**
  - Applied research activities, analytic studies, small code units, and papers comparing competing technologies

DS-TRL-3

- **Definition**
  - Analytical and experimental critical function and/or characteristic proof of concept.
- **Description**
  - Active R&D is initiated. The level at which scientific feasibility is demonstrated through analytical and laboratory studies. This level extends to the development of limited functionality environments to validate critical properties and analytical predictions using nonintegrated software components and partially representative data.
- **Supporting Information**
  - Algorithms run on a surrogate processor in a laboratory environment, instrumented components operating in laboratory environment, laboratory results showing validation of critical properties.

DS-TRL-4

- **Definition**
  - Module and/or subsystem validation in a laboratory environment (i.e., software prototype development environment)
- **Description**
  - Basic software components are integrated to establish that they will work together. They are relatively primitive with regard to efficiency and robustness compared with the eventual system. Architecture development initiated to include interoperability, reliability, maintainability, extensibility, scalability, and security issues. Emulation with current/ legacy elements as appropriate. Prototypes developed to demonstrate different aspects of eventual system.
- **Supporting Information**
  - Advanced technology development, stand-alone prototype solving a synthetic full-scale problem, or standalone prototype processing fully representative data sets.

DS-TRL-5

- **Definition**
  - Module and/or subsystem validation in a relevant environment.
- **Description**
  - Level at which software technology is ready to start integration with existing systems. The prototype implementations conform to target environment/ interfaces. Experiments with realistic problems. Simulated interfaces to existing systems. System software architecture established. Algorithms run on a processor(s) with characteristics expected in the operational environment.
- **Supporting Information**
  - System architecture diagram around technology element with critical performance requirements defined. Processor selection analysis, Simulation/ Stimulation (Sim/Stim) Laboratory buildup plan. Software placed under configuration management. COTS/GOTS in the system software architecture are identified.

DS-TRL-6

- **Definition**
  - Module and/or subsystem validation in a relevant end-to-end environment.
- **Description**
  - Level at which the engineering feasibility of a software technology is demonstrated. This level extends to laboratory prototype implementations on full-scale realistic problems in which the software technology is partially integrated with existing hardware/ software systems.
- **Supporting Information**
  - Results from laboratory testing of a prototype package that is near the desired configuration in terms of performance, including physical, logical, data, and security interfaces. Comparisons between tested environment and operational environment analytically understood. Analysis and test measurements quantifying contribution to system-wide requirements such as throughput, scalability, and reliability. Analysis of human-computer (user environment) begun.

DS-TRL-7

- **Definition**
  - System prototype demonstration in an operational high-fidelity environment
- **Description**
  - Level at which the program feasibility of a software technology is demonstrated. This level extends to operational environment prototype implementations where critical technical risk functionality is available for demonstration and a test in which the software technology is well integrated with operational hardware/software systems.
- **Supporting Information**
  - .Critical technological properties are measured against requirements in a simulated operational environment.

DS-TRL-8

- **Definition**
  - Actual system completed and mission qualified through test and demonstration in an operational environment.
- **Description**
  - Level at which a software technology is fully integrated with operational hardware and software systems. Software development documentation is complete. All functionality tested in simulated and operational scenarios.
- **Supporting Information**
  - Published documentation and product technology refresh build schedule. Software resource reserve measured and tracked.

DS-TRL-9

- **Definition**
  - Actual system proven through successful mission- proven operational capabilities.
- **Description**
  - Level at which a software technology is readily repeatable and reusable. The software based on the technology is fully integrated with operational hardware/software systems. All software documentation verified. Successful operational experience. Sustaining software engineering support in place. Actual system.
- **Supporting Information**
  - Production configuration management reports. Technology integrated into a reuse "wizard"; out-year funding established for support activity

Improving Systems Software Development: *Government Responsibility*

- **Start earlier with systems software and tie it to the KDPs**
- **Create a documented viable Government Reference Integrated Hardware/Software Architecture, also known as the Government Reference Architecture (GRA)**
- **As preparation for the GRA, initial concept studies should be undertaken as needed in a facility such as The Aerospace Corporation's Concept Design Center (CDC)**
- **Include plans for a simulation environment in the GRA**

Improving Software Development: *Government Documents*

- **Ensure a Concept of Operations (CONOPS) and Operational Concept Documents (OCD) have been created**
- **Assure the proper state of developmental documents including a System Subsystem Design Description (SSDD) and Software Development Plan (SDP)**
- **Define a common system terminology dictionary with the stakeholders**

Improving Software Development: *Contractor Responsibilities*

- **Assure that the contractor is developing and utilizing prototyping and simulations to demonstrate readiness.**
- **Assure that there is a specific management plan for handling all Non-Developmental Item (NDI) software and its version upgrades and governance issues**
- **Assure proper identification of and compliance testing of Commercial-off-the shelf software (COTS)/government off the shelf (GOTS) in the system software architecture related to the SS-CTE**

Improving Software Development: *Contractor Documents*

- **Assure that the contractor is developing and utilizing prototyping and simulations to demonstrate readiness.**
- **Assure that there is a specific management plan for handling all Non-Developmental Item (NDI) software and its version upgrades and governance issues**
- **Assure proper identification of and compliance testing of Commercial-off-the shelf software (COTS)/government off the shelf (GOTS) in the system software architecture related to the SS-CTE**

References

1 Chaplain, C., GAO-07-1088R Space Based Infrared System High Program and its Alternative, September 2007, http://www.gao.gov/new.items/d071088r.pdf

2 Department of Defense Technology Readiness Assessment (TRA) Deskbook, May 2005, https://acc.dau.mil/CommunityBrowser.aspx?id=23349

3 Adams, R., Software Acquisition Best Practices: Experiences from the Space Systems Domain

September 2004

4 Eslinger, S., Software Acquisition Best Practices Tutorial, September 2005

5 Adams, R., Reducing Software Acquisition Risk: Best Practices for the Early Acquisition Process

January 2006

6 National Security Space Acquisition Policy Number 03-01, 27 December 2004 https://akss.dau.mil/Documents/Policy/Space%20Acquisition%20NSSAcqPol0301_signed_%2027Dec04%20(GN).pdf

7 Gold, R., Technology Readiness Assessments for IT and IT-Enabled Systems, May 2005

8 Software Program Managers Network Lessons Learned Page

http://www.spmn.com/lessons.html

9 Rechtin, E., Systems Architecting: Creating and Building Complex Systems, 1991, Prentice Hall

References (continued)

10 Joint Program Management Handbook, July 2004, Defense Acquisition University Press http://www.dau.mil/pubs/misc/Joint%20PM%20Handbook%2010_2004.pdf

11 Hantos, P., Software Technology Readiness Assessments – Managing Software Technology Risks in Space System Acquisitions, February 2008

12 Aguilar, J., Scope vs. Detail: The Teams of the Concept Design Center, December 1999, http://ieeexplore.ieee.org/iel5/7042/19022/00879431.pdf?arnumber=879431

13 Florence, A., Requirements for a Chief Software Engineer for a Weapons System Acquisition January 2004 http://www.sei.cmu.edu/programs/acquisition-support/conf/2004-presentations/florence.pdf

14 Ask a Professor: What is the difference between an Operational Concept (OPCON) document and a Concept of Operations (CONOPS)?, March 2005 https://akss.dau.mil/askaprof-akss/qdetail2.aspx?cgiSubjectAreaID=14&cgiQuestionID=14094

15 Young, J., Prototyping and Competition Policy Memorandum September 2007 https://acc.dau.mil/CommunityBrowser.aspx?id=170517&lang=en-US

16 Common Procurement Vocabulary, December 2003 http://europa.eu/scadplus/leg/en/lvb/l22008.htm

17 Federal Aviation Administration Acquisition Management System 4.1.8 : Commercial Off-The-Shelf, Non-Developmental Items, and Commercially Available Software, February 2004 http://fasteditapp.faa.gov/ams/do_action?do_action=LinkSection&contentUID=1&sectionNumber=4#FAA_330