Systems & Software Technology Conference

Improving Software Guidance for Technology Readiness Assessments (TRA)

29 April 2010

Mike Nicol
Software Technical Advisor
Air Force Aeronautical Systems Center
Engineering Directorate (ASC/EN)

Integrity - Service - Excellence

U.S. AIR FORCE

Outline

- Technology Maturity Requirements
- Issues & Considerations
- Initial Recommendations
- Considerations for Software TRL Definitions
- Closing Thoughts

Technology Maturity Requirements

- **Title 10 U.S. Code, Section 2366b:**
  - **"A major defense acquisition program may not receive Milestone B approval ... until the milestone decision authority certifies that ... the technology in the program has been demonstrated in a relevant environment"**
- **DoDI 5000.02, December 2, 2008, Entrance Criteria for EMD:**
  - **"Entrance into this phase depends on technology maturity (including software)…**
- **Directive Type Memorandum (DTM) 09-027, December 4, 2009**
  - **MDA is required to certify that "the technology in the program has been demonstrated in a relevant environment, as determined by the Milestone Decision Authority on the basis of an independent review and assessment by the Director of Defense Research and Engineering"**

Technology Maturity Requirements (Cont.)

U.S. AIR FORCE

- All Department Of Defense (DoD) Acquisition Programs Must Have A Formal TRA At Milestone B And At Milestone C Of The Defense Acquisition System
- TRAs For Acquisition Category (ACAT) ID And IAM Programs Must Be Submitted To The Director, Research Directorate (DRD) In The Office Of The Director Of Defense Research And Engineering (DDR&E)
- MDA Must Certify That The Technology In Major Defense Acquisition Programs (MDAPS)
  - Has been demonstrated in a relevant environment (TRL 6) before Milestone B approval
  - Is at a state of technology maturity of TRL 7 or higher at Milestone C
    - "All software has been written and tested, not only as an independent module and/or component, but also as integrated into the whole system…"

U.S. AIR FORCE

Definitions

- **Software Technology:** Software technology is defined as the theory and practice of various sciences applied to software development, operation, understanding, and maintenance. Software Technology is any concept, process, method, algorithm, or tool whose primary purpose is the development, operation, understanding, and maintenance of software. [Foreman, 1997]
- **Algorithm:** In mathematics, computer science, and related subjects, an algorithm is an effective method for solving a problem expressed as a finite sequence of instructions. [Wikipedia]
- **Critical Technology Element (CTE):** A technology element is "critical" if the system being acquired depends on this technology element to meet operational requirements (within acceptable cost and schedule limits) and if the technology element or its application is either new or novel or in an area that poses major technological risk during detailed design or demonstration. [TRA Deskbook]

What Is A CTE?

U.S. AIR FORCE

The Process Of Developing CTE Candidates Relies On A Series Of Questions To Test Whether The CTE Definition Applies:

1. **Does the technology have a significant impact on an operational requirement, cost, or schedule?**
2. **Does this technology pose a major development or demonstration risk?**
3. **Is the technology new or novel?**
4. **Has the technology been modified from prior successful use?**
5. **Has the technology been repackaged such that a new relevant environment is applicable?**
6. **Is the technology expected to operate in an environment and/or achieve a performance beyond its original design intention or demonstrated capability?**

The First Test To Be Passed Is Whether The Technology Is Critical, As Determined By A "Yes" Answer To Question 1. The Second Test Is Whether Any Of The Remaining Questions Can Be Answered With A "Yes." If So, Then The Technology Is A CTE.

[TRA Deskbook, Appendix B]

Issues & Considerations

My Experience With Software TRAs

- **Lack Of Clear Software Guidance Prolongs The CTE Identification Process**
- **TRA Independent Review Teams (IRTs) Are Not Comfortable With Identifying No Software CTEs**
- **Software Product Issues That Would Be Better Addressed As Program Risks Are Identified As CTEs**

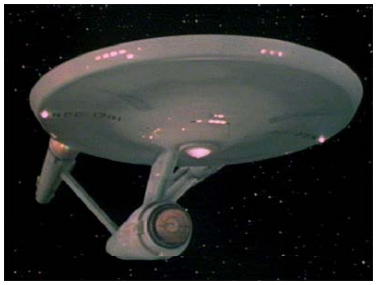
Issues & Considerations

"Software CTE" Candidates?

- **Development Tools Used In A Manner Different From Prior Uses?**

- **85% Planned Software Reuse In A System With 5 Million Lines Of Code?**

- **Transporter Software For The First Starship Enterprise?**



Issues & Considerations

Specifics

- **TRL 6 At MS-B Approval (Demonstrated In A Relevant {Simulated} Environment) Is Problematic For Software That Has Not Yet Been Developed**
  - **“Mature software” at MS-B implies Non Developmental Items (NDI) and assumes fully integrated and tested**
  - **Cannot assess the maturity of software not yet developed**
  - **Complete understanding of the underlying technology (physics, algorithms, equations, etc.) is required before software requirements can be stated**
  - **Software life cycle is not the same as DoDI 5000.02 system life cycle**

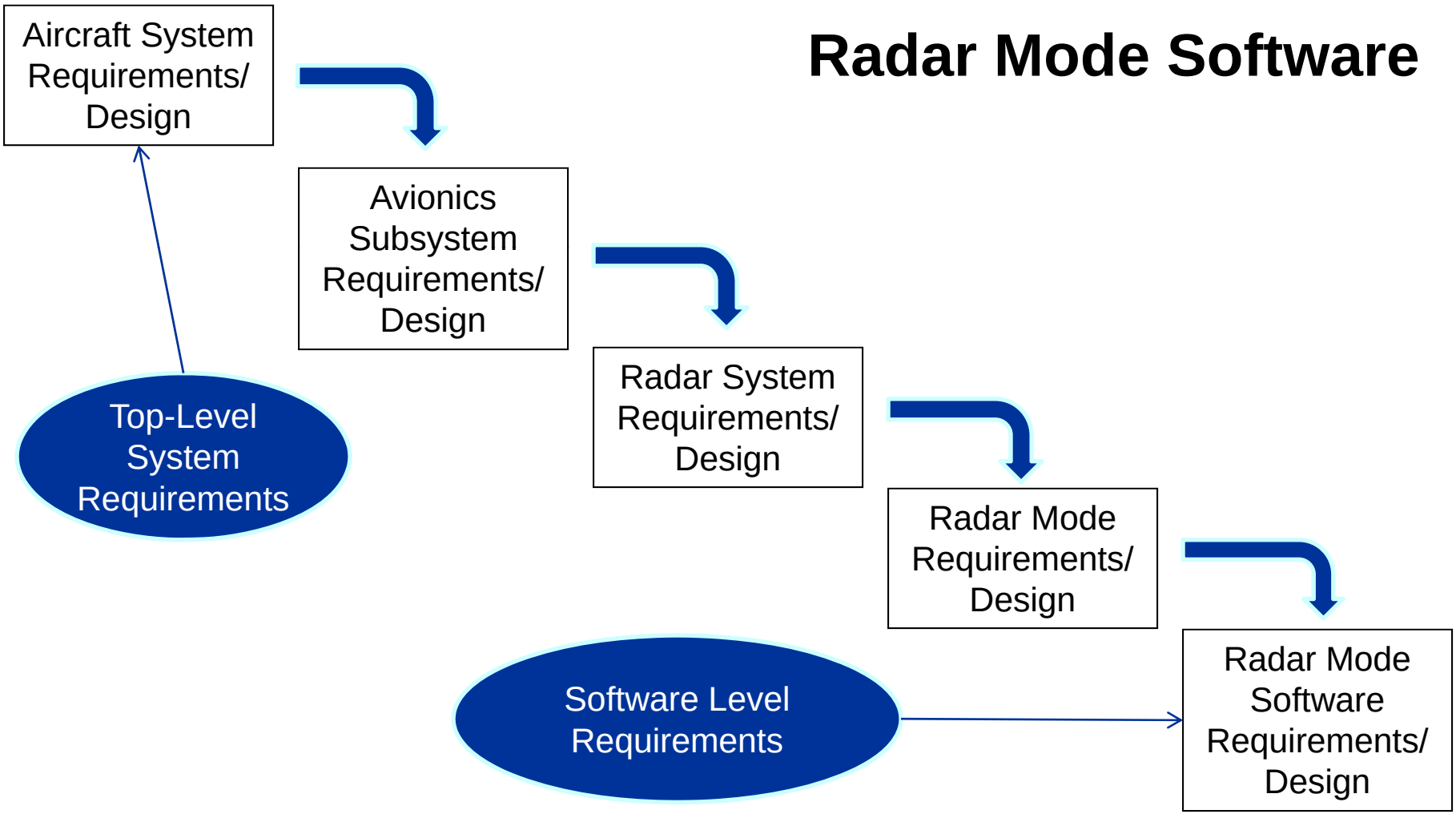
Issues & Considerations

DoDI 5000.02 PDR Requirement

- **DoDI 5000.02, 2 Dec 2008:**

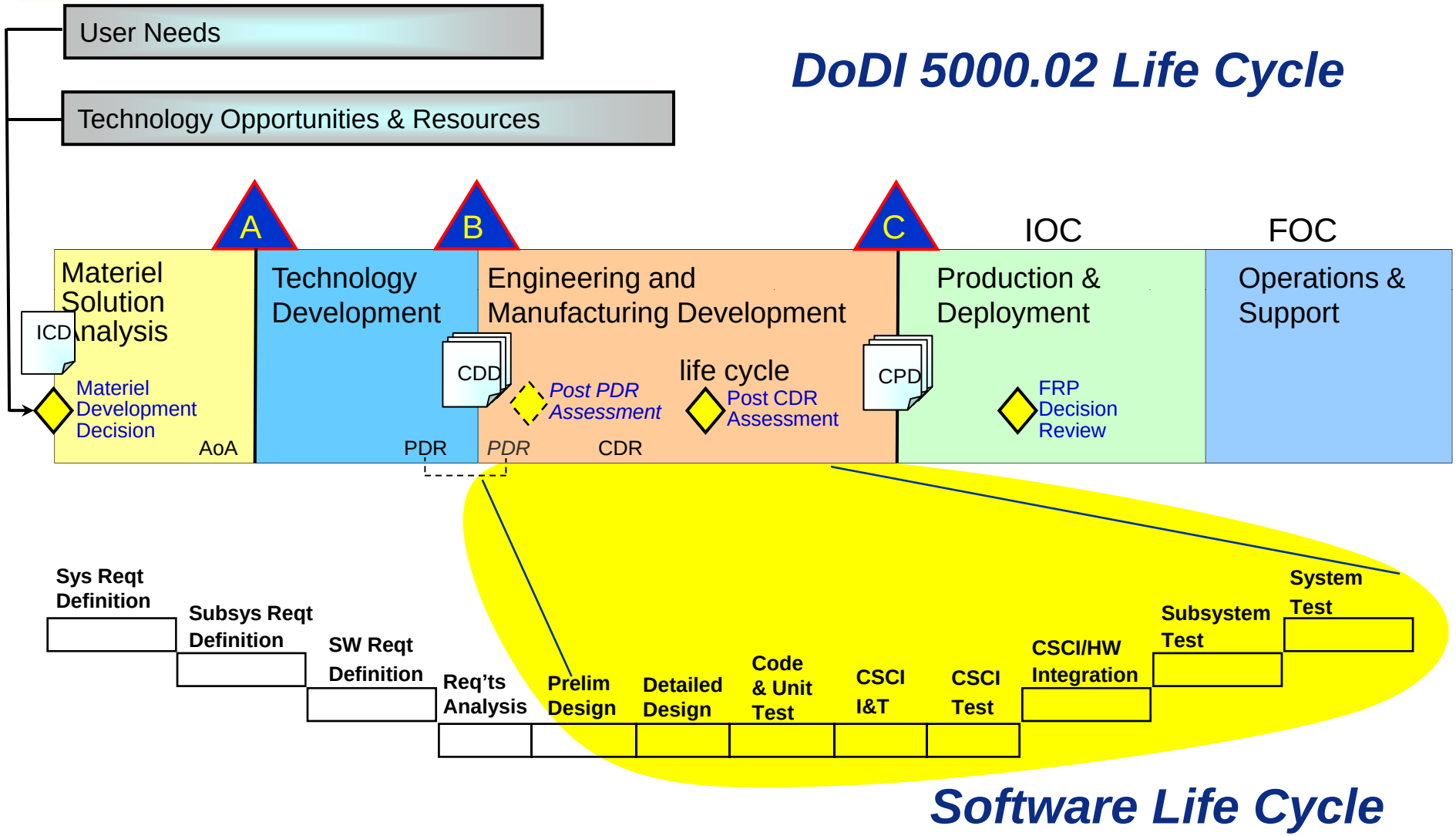
  5.d (6): When consistent with Technology Development Phase objectives, associated prototyping activity, and the MDA approved TDS, the PM shall plan a Preliminary Design Review (PDR) before Milestone B. PDR planning shall be reflected in the TDS and shall be conducted for the candidate design(s) to establish the allocated baseline (hardware, *software*, human/support systems) and underlying architectures and to define a high-confidence design. All system elements (hardware and *software*) shall be at a level of maturity commensurate with the PDR entrance and exit criteria. *A successful PDR will inform requirements trades; improve cost estimation; and identify remaining design, integration, and manufacturing risks.* The PDR *shall be conducted at the system level* and include user representatives and associated certification authorities. The PDR Report shall be provided to the MDA at Milestone B and include recommended requirements trades based upon an assessment of *cost, schedule, and performance risk*.

Software Design Evolution Example



Technology Vs. Software Lifecycle

U.S. AIR FORCE



Issues & Considerations

Specifics (Cont.)

- **What Is A Software CTE?**
- **What is the Expectation for COTS/NDI Vs. New Development Software?**
  - **Is there an assumed TRL associated with existing COTS/NDI software?**
  - **How should the significant challenges of testing COTS/NDI in a relevant environment be addressed?**
  - **Assuming that integration is the biggest risk in large applications of COTS/NDI, how should post MS-B integration activities be addressed?**

Issues & Considerations

Specifics (Cont.)

- **How Do Programs Separate Routine Software-Related Program Risks From Software Critical Technology Elements (CTEs)?**
- **What Is A Relevant Environment For Software?**
- **What Is A Prototype For Developed Software?**
- **Are The Existing Software TRL Definitions Adequate/Appropriate?**

U.S. AIR FORCE

Initial Recommendations

- **Improve Language In TRA Deskbook To Clarify Intent For Weapon System Software**
  - **Separate program risks and software product from "technology maturity"**
  - **Separate software product from software technology**
  - **Address various software types: new development and NDI (COTS, GOTS, OSS, and other reuse)**
  - **Address various domains: embedded software in weapon systems, functional & ERP systems, etc.**

Initial Recommendations (Cont.)

U.S. AIR FORCE

- Revise Software TRL Definitions To Make Them More Consistent With The System/Software Development Life Cycle
- Consider A More Integrated Approach To Program Evaluation That Can Identify And Differentiate Between Technology And Programmatic Risks
- Define/Clarify "Relevant Demonstrated Environment For Software"
- Provide Guidance On Dealing With Integration, System Of Systems, Etc.
- Provide Guidance On TRA/TRL Best Practices And Tools To Measure CTE Progress Toward The Next Level Of Maturity

Initial Recommendations (Cont.)

- **Develop Balanced Guidance And Training For Software TRAs**
- **Reconsider TD-1-12 (Apr 09) Report Recommendations For TRA Deskbook**
  - **Hantos, Peter, and TD-1-12 Software Sub-Team: *Software Technology Readiness Assessment Recommendations*, Air Force Smart Operations – 21 Developing & Sustaining Warfighting Systems, 2009**

Considerations for Software TRL Definitions

Non-Technology Elements of Current Software TRL Definitions

U.S. AIR FORCE

- **TRL 5, Supporting Information: "Software placed under configuration management."**
- **TRL 8, Description: "Software development documentation is complete."**
- **TRL 8, Supporting Information: "Published documentation and product technology refresh build schedule. Software resource reserve measured and tracked."**
- **TRL 9, Description: "All software documentation verified. … Sustaining software engineering support in place."**
- **TRL 9, Supporting Information: "Production configuration management reports. Technology integrated into a reuse 'wizard'."**

Considerations for Software TRL 1

U.S. AIR FORCE

Is:	Description	Supporting Information
1 - Basic Principles Observed and Supported	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.	Basic research activities, research articles, peer-reviewed white papers, point papers, early lab model of basic concept may be useful for substantiating the TRL.
Proposed:	**Description**	**Supporting Information**
1 - System Needs	TBD	TBD

Considerations for Software TRL 2

U.S. AIR FORCE

Is:	Description	Supporting Information
2 - Technology concept and/or application formulated	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.	Applied research activities, analytic studies, small code units, and papers comparing competing technologies.
Proposed:	**Description**	**Supporting Information**
2 - Operational Concept	TBD	TBD

Considerations for Software TRL 3

U.S. AIR FORCE

Is:	Description	Supporting Information
3 - Analytical and experimental critical function and/or characteristic proof of concept	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 non-integrated software components and partially representative data.	Algorithms run on a surrogate processor in a laboratory environment, instrumented components operating in a laboratory environment, laboratory results showing validation of critical properties.
Proposed:	**Description**	**Supporting Information**
3 - System Requirements	System specification level requirements for computer systems and software are complete.	TBD

Considerations for Software TRL 4

U.S. AIR FORCE

Is:	Description	Supporting Information
4 - Module and/or subsystem validation in a laboratory environment (i.e., software prototype development environment)	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.	Advanced technology development, stand-alone prototype solving a synthetic full-scale problem, or standalone prototype processing fully representative data sets.
Proposed:	**Description**	**Supporting Information**
4 - System Design	System designs are complete. Subsystem requirements and architectures are defined. Initial computer system and software architectures are defined.	TBD

Considerations for Software TRL 5

U.S. AIR FORCE

Is:	Description	Supporting Information
5 - Module and/or subsystem validation in a relevant environment	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.	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. Commercial-off-the-shelf/government-off-the-shelf (COTS/GOTS) components in the system software architecture are identified.
Proposed:	**Description**	**Supporting Information**
5 - Subsystem Design	Subsystem designs are complete and subsystem functions are allocated to hardware and software. Software and interface requirements specifications are complete.	TBD

Considerations for Software TRL 6

(Milestone B Requirement)

U.S. AIR FORCE

Is:	Description	Supporting Information
6 - Module and/or subsystem validation in a relevant end-to-end environment	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.	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.
Proposed:	**Description**	**Supporting Information**
6 - Software architecture and requirements defined	Software structure is established, components and relationships are identified, and interfaces are defined. *Risks are identified. Software size, effort, and schedule are estimated.*	TBD

Considerations for Software TRL 7

(Milestone C Target)

U.S. AIR FORCE

Is:	Description	Supporting Information
7 - System prototype demonstration in an operational high-fidelity environment	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.	Critical technological properties are measured against requirements in an operational environment.
Proposed:	**Description**	**Supporting Information**
7 - System/Software performance is verified in the lab and operational testing to date is satisfactory	TBD	TBD

Considerations for Software TRL 8

(Milestone C Preferred)

U.S. AIR FORCE

Is:	Description	Supporting Information
8 - Actual system completed and mission qualified through test and demonstration in an operational environment	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.	Published documentation and product technology refresh build schedule. Software resource reserve measured and tracked.
Proposed:	**Description**	**Supporting Information**
8 - Operational testing verifies system/software are safe, suitable, and effective	TBD	TBD

Considerations for Software TRL 9

U.S. AIR FORCE

Is:	Description	Supporting Information
9 - Actual system proven through successful mission-proven operational capabilities	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.	Production configuration management reports. Technology integrated into a reuse "wizard."
Proposed:	**Description**	**Supporting Information**
9 - System & software proven in operational use	TBD	TBD

U.S. AIR FORCE

Closing Thoughts

Goals for Program Starts

- **How To Establish High Confidence Programs With Low To Moderate Risk?**
- **Address All Software-Related Concerns**
  - Software sources (new development, COTS, GOTS, open source, other reuse)
  - Technical solution (compliance with requirements, soundness of approach, architecture, safety/mission criticality, **maturity**)
  - Developer capability & capacity (domain experience, organization, staffing, processes)
  - Infrastructure (development tools, integration labs, etc.)
  - Compatibility of software size, effort, and schedule with program technical, cost, and schedule baselines

Way Ahead

- Work With Other Services And OSD To Recommend And Implement Improvements
- Air Force Contacts
  - Mike Nicol, ASC/EN
    - Michael.Nicol@wpafb.af.mil
    - 937-255-9566
  - Ernesto Gonzalez, (SAF/AQRE)
    - Ernesto.Gonzalez.ctr@pentagon.af.mil
    - 703-254-2474

Backups

TRA Deskbook Software TRL Definitions, Descriptions, and Supporting Information

U.S. AIR FORCE

	Description	Supporting information
1 - Basic Principles Observed and Supported	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.	Basic research activities, research articles, peer-reviewed white papers, point papers, early lab model of basic concept may be useful for substantiating the TRL.
2 - Technology concept and/or application formulated	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.	Applied research activities, analytic studies, small code units, and papers comparing competing technologies.
3 - Analytical and experimental critical function and/or characteristic proof of concept	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 non-integrated software components and partially representative data.	Algorithms run on a surrogate processor in a laboratory environment, instrumented components operating in a laboratory environment, laboratory results showing validation of critical properties.
4 - Module and/or subsystem validation in a laboratory environment (i.e., software prototype development environment)	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.	Advanced technology development, stand-alone prototype solving a synthetic full-scale problem, or standalone prototype processing fully representative data sets.
5 - Module and/or subsystem validation in a relevant environment	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.	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. Commercial-off-the-shelf/government-off-the-shelf (COTS/GOTS) components in the system software architecture are identified.
6 - Module and/or subsystem validation in a relevant end-to-end environment	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.	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.
7 - System prototype demonstration in an operational high-fidelity environment	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.	Critical technological properties are measured against requirements in an operational environment.
8 - Actual system completed and mission qualified through test and demonstration in an operational environment	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.	Published documentation and product technology refresh build schedule. Software resource reserve measured and tracked.
9 - Actual system proven through successful mission-proven operational capabilities	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.	Production configuration management reports. Technology integrated into a reuse "wizard."

TRA Deskbook Hardware TRL Definitions, Descriptions, and Supporting Information

U.S. AIR FORCE

	Description	Supporting information
1 - Basic principles observed and reported.	Lowest level of technology readiness. Scientific research begins to be translated into applied research and development (R&D). Examples might include paper studies of a technology's basic properties.	Published research that identifies the principles that underlie this technology. References to who, where, when.
2 – Technology concept and/or application formulated.	Invention begins. 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.	Publications or other references that outline the application being considered and that provide analysis to support the concept.
3 - Analytical and experimental critical function and/or characteristic proof of concept.	Active R&D is initiated. This includes analytical studies and laboratory studies to physically validate the analytical predictions of separate elements of the technology. Examples include components that are not yet integrated or representative.	Results of laboratory tests performed to measure parameters of interest and comparison to analytical predictions for critical subsystems. References to who, where, and when these tests and comparisons were performed.
4 – Component and/or breadboard Validation in a laboratory environment.	Basic technological components are integrated to establish that they will work together. This is relatively "low fidelity" compared with the eventual system. Examples include integration of "ad hoc" hardware in the laboratory.	System concepts that have been considered and results from testing laboratory-scale breadboard(s). References to who did this work and when. Provide an estimate of how breadboard hardware and test results differ from the expected system goals.
5 - Component and/ or breadboard validation in a Relevant environment.	Fidelity of breadboard technology increases significantly. The basic technological components are integrated with reasonably realistic supporting elements so they can be tested in a simulated environment. Examples include "high-fidelity" laboratory integration of components.	Results from testing a laboratory breadboard system are integrated with other supporting elements in a simulated operational environment. How does the "relevant environment" differ from the expected operational environment? How do the test results compare with expectations? What problems, if any, were encountered? Was the breadboard system refined to more nearly match the expected system goals?
6 - System/subsystem model or prototype demonstration in a Relevant environment.	Representative model or prototype system, which is well beyond that of TRL 5, is tested in a 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.	Results from laboratory testing of a prototype system that is near the desired configuration in terms of performance, weight, and volume. How did the test environment differ from the operational environment? Who performed the tests? How did the test compare with expectations? What problems, if any, were encountered? What are/were the plans, options, or actions to resolve problems before moving to the next level?
7 - System prototype demonstration in an operational environment.	Prototype near or at planned operational system. Represents a major step up from TRL 6 by requiring demonstration of an actual system prototype in an operational environment (e.g., in an aircraft, in a vehicle, or in space).	Results from testing a prototype system in an operational environment. Who performed the tests? How did the test compare with expectations? What problems, if any, were encountered? What are/were the plans, options, or actions to resolve problems before moving to the next level?
8 - Actual system completed and qualified through test and demonstration.	Technology has been proven to work in its final form and under expected conditions. In almost all cases, this TRL represents the end of true system development. Examples include developmental test and evaluation (DT&E) of the system in its intended weapon system to determine if it meets design specifications.	Results of testing the system in its final configuration under the expected range of environmental conditions in which it will be expected to operate. Assessment of whether it will meet its operational requirements. What problems, if any, were encountered? What are/were the plans, options, or actions to resolve problems before finalizing the design?
9 - Actual system proven through successful mission operations.	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.	OT&E reports.