A Software Technology Readiness Assessment Process for Minimizing System Acquisition Risk

Willie J. Fitzpatrick, PhD
Chief, Aviation Division
SED, AMRDEC
256-876-9945

November 17, 2005

Purpose

Present the Software Engineering Directorate (SED) Software Technology Readiness Assessment Process.

Background

DoD have set policies for implementing Technology Readiness Assessment (TRA):

- DoD Directive 5000.1, *The Defense Acquisition System*, *May 2003*, Specifically calls for software to be included in TRA.
- DoD Instruction 5000.2, *Operation of the Defense Acquisition System*, *May 2003*, provides the process for performing TRA.
- DoD *Technology Readiness Assessment (TRA) Deskbook*, *May 2005* provides guidance for System and Software definitions of technology maturity; **but no detailed maturity criteria.**

System Acquisition Framework

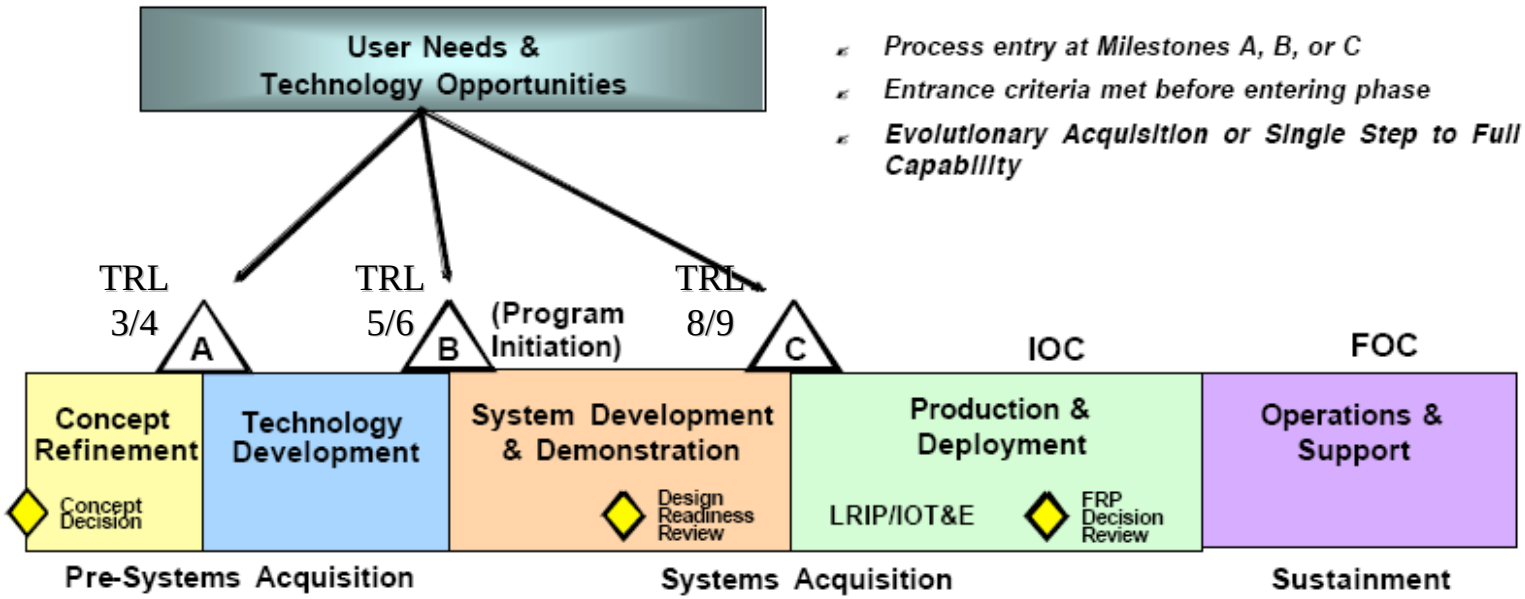

Figure 2-1. Defense Acquisition Management Framework
(Source: DoDI 5000.2)

Technology Readiness Levels

- The TRA process results in the Technology Readiness Level (TRL) metric for technologies to be implemented or to characterize an existing system
- The TRL is a set of definitions for 9 levels of technology maturity:
  - **TRL 1 being the least mature.**
  - **TRL 9 being the most mature.**

System Technology Readiness Levels (1of 2)

1. Basic principles observed and reported.	Lowest level of technology readiness. Scientific research begins to be translated into applied research and development. Examples might include paper studies of technology's basic properties.
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.
3. Analytical and experimental critical function and/or characteristic proof of concept.	Active research and development is initiated. This includes analytical studies and laboratory studies to physically validate analytical predictions of separate elements of the technology. Examples include components that are not yet integrated or representative.
4. Component and/or breadboard validation in laboratory environment.	Basic technological components are integrated to establish that they will work together. This is relatively "low fidelity" compared to the eventual system. Examples include integration of "ad hoc" hardware in a laboratory.
5. Component and/or breadboard validation in relevant environment.	Fidelity of breadboard technology increases significantly. The basic technological components are integrated with reasonably realistic supporting elements so it can be tested in simulated environment. Examples include "high fidelity" laboratory integration of components.

System Technology Readiness Levels (2of 2)

6. System/subsystem model or prototype demonstration in a relevant environment.	Representative model or prototype system, which is well beyond the breadboard tested for level 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 simulated operational environment.
7. System prototype demonstration in an operational environment.	Prototype near, or at, planned operational system. Represents a major step up from TRL 6, requiring demonstration of an actual system prototype in an operational environment such as an aircraft, vehicle, or space. Examples include testing the prototype in a test bed aircraft.
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 level represents the end of true system development. Examples include developmental test and evaluation of the system in its intended weapon system to determine if it meets design specifications.
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. Examples include using the system under operational mission conditions.

Software Technology Readiness Assessment Process

Problem

- How to equate software TRLs to other system element TRLs, e.g., Hardware, Operations, Mission Plan, & Intelligence?

and

- How to relate software TRL's to Software Risk?

Software Technology Readiness Assessment Process

Solution

- Define and use software TRLs consistent with the system TRLs.
- Perform Technology Assessment procedures for software (use software TRL definitions).
- Correlate software TRLs to Risk Assessment.

System Acquisition Framework

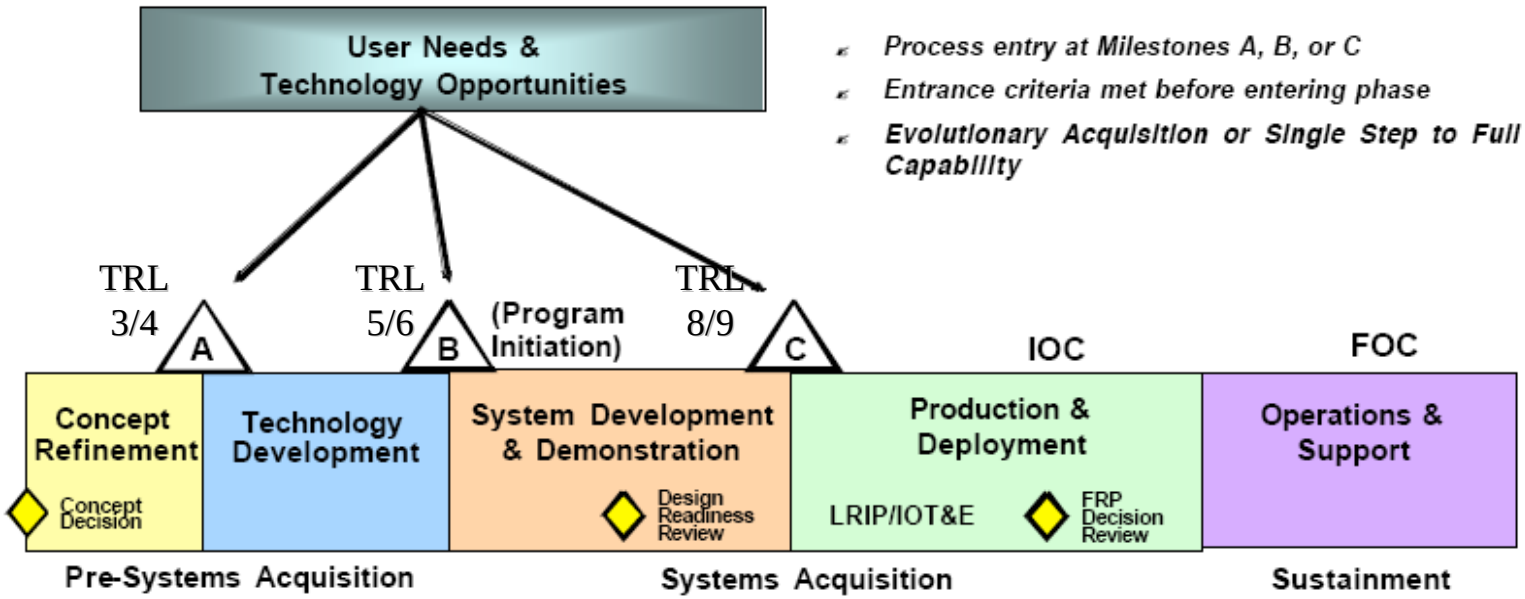

Figure 2-1. Defense Acquisition Management Framework
(Source: DoDI 5000.2)

Software Technology Readiness Levels

Table 3-3. Software TRL Definitions, Descriptions, and Supporting Information
(Source: *IT TRL Working Group Minutes*, November 9, 2004)

TRL	Definition	Description	Supporting Information
1	Basic principles observed and reported.	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 level.
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 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.

Software Technology Readiness Levels

Table 3-3. Software TRL Definitions, Descriptions, and Supporting Information (Source: *IT TRL Working Group minutes*, November 9, 2004) (Continued)

TRL	Definition	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. COTS/GOTS 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 a simulated operational environment.

Software Technology Readiness Levels

Table 3-3. Software TRL Definitions, Descriptions, and Supporting Information (Source: *IT TRL Working Group Minutes*, November 9, 2004) (Continued)

TRL	Definition	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.
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"; out-year funding established for support activity.

Criteria for Software TRL

Maturity Status Indicators →	**Previous System Documents/ Code Availability**	**Technology Prototyped/ Used Existing System**	**Studies/Test Use Results**	**Open Problem Reports**	**Test (Verify) Environment**	**Develop-ment Process**	**Safety/ Security**	**Precision/ Perform-ance**	**Develop-ment Environ-ment**	**Change To Code**
TRL 1	Concept Documented	Concept Documented	Ops. Need Documented	N/A	N/A	N/A	N/A	N/A	N/A	N/A
TRL 2	Design Approach Analysis & Feasibility Doc.	Studies/ Simulation/ Design Documented	• Ops. Use Likely • Possible Application Identified	N/A	N/A	N/A	N/A	N/A	N/A	N/A
TRL 3	Preliminary Design Doc.	Prototype Approach & Plan Doc.	• Ops. Use Possible • Cost/Benefit Doc.	N/A	N/A	N/A	N/A	N/A	N/A	N/A
TRL 4	Prototype Code/Doc.	Technology Prototyped/ Simulated	• Suitable for Ops. Use • Candidate Systems Identified	Fixes For Observed Problems Feasible	Host Emulator or Breadboard	Design Doc. & Code in Library	N/A	N/A	N/A	N/A
TRL 5	Existing System/ Prototype Code/Doc.	Technology Prototyped/ Existing System	• Requirements/ Algorithms Ok for Ops. Use • Cost/Benefits Updated	Fixes for Reported Problems Feasible & Within Dev. Plan	Host Emulator or Typical Target/ Breadboard	CM & Problem Process Doc. & Used	Existing System/ Prototype Suitable with Changes	Existing System/ Prototype Suitable with Changes	COTS	N/A

Criteria for Software TRL (Cont'd)

Maturity Status Indicators →	**Previous System Documents/ Code Availability**	**Technology Prototype/ Used Existing System**	**Studies/Test Use Results**	**Open Problem Reports**	**Test (Verify) Environment**	**Develop-ment Process**	**Safety/ Security**	**Precision/ Perform-ance**	**Develo-ment Envirn-ment**	**Change To Code**
TRL 6	TRL 5 Code & Doc	Prototype/ Existing System Used in Typical Ops Target/ Brdbrd.	- Requirements/ Algorithms/ Design Ok for Ops Use - Some Req/Design Changes Ok	Fixes for Reported Problems Feasible & Within Dev Plan	- Typical Ops Target/ Simulator/ Brdbrd. - Canned Ops Data	CMM Level 2 or Higher Doc. & Followed	System Suitable With Low Risk Changes	System Suitable But May Need Tuning	COTS	Change to TRL 5 Code < 10.0%
TRL 7	TRL 6 Code & Doc	TRL 6 Software Used in Ops Fidelity Target/ Brdbrd.	- Requirements/ Algorithms/ Design Ok for Ops Use - Some Design Changes Ok	- No Open Priority 1 Problems - Problem Fixes Within Dev Plan	- Typical Ops. Target/ Simulator/ Brdbrd. - Dynamic Ops Data	CMM Level 3 or Higher Followed	System Suitable With Minor Low-risk Changes	- System Suitable -No Changes to Req. or Design	COTS	Change to TRL 6 Code < 5.0%
TRL 8	TRL 7 Code & Doc	TRL 7 Software Field Tested or Used in System IOT&E	Acceptable with Only Minor Design Changes	No Open Priority 1, 2, 3 Problems	Ops Fidelity System or Field Test on Target	CMM Level 3 or Higher Followed	No Changes to Req. or Design	- System Suitable - Minor Changes to Code	COTS Same as Used for TRL 7	Change to TRL 7 Code < 1.0%
TRL 9	TRL 8 Code & Doc	TRL 8 Software Field Tested or Used in System IOT&E	Acceptable with Only Changes to the Algorithm Parameters	No Open Priority 1, 2, 3 Problems	Ops Fidelity System or Field Test on Target	CMM Level 3 or Higher Followed	No Changes to Code	- System Suitable - No Code Changes	COTS Same as Used for TRL 8	Change to TRL 8 Code < 0.5%

Software TRL for the Systems Documents/Code Availability Maturity Status Indicator

- Purpose: to determine if there is sufficient documentation to perform inspections, safety, security analyses, and efficient and reliable maintenance for FQT and post release support.

- TRL 9 Criterion: all of the pertinent documents and data specified in J-STD-16/DOD STD-490/IEEE 1228 have been produced, are current, and have been verified as correct and complete.

Software TRL for the Technology Prototyped/Operational Testing/Use Maturity Status Indicator

- Purpose: to determine the state of the technology operational experience.

- TRL 9 Criterion: the technology has been used in a similar operational application and system configuration (e.g., deployed system, field tested system, successfully used in system level testing/IOT&E).

Software TRL for the Operational Suitability

Maturity Status Indicator

- Purpose: to determine the degree to which the operational functional requirements (e.g., logic, algorithms/design) meet operational functional requirements.

- TRL 9 Criterion: the technology meets the system operational functional requirements with only changes to the algorithm parameters for the system under consideration.

Software TRL for the Problem Report Status Maturity Status Indicator

- Purpose: to determine the significance of the problem reports generated after the software code has been baselined (put under CM in the Software Development Library/code to be used for the FQT and code to be released after FQT).

- TRL 9 Criterion: the technology code has a demonstrated reliability as a result of formal testing consistent with no unresolved DOD priority 1 (catastrophic), 2 (severe), and 3 (moderate) reported problems.

Software TRL for the
Test (Verify) Environment
Maturity Status Indicator

- Purpose: to determine that the test environment (e.g., simulators, emulators, breadboards, target prototypes, and targets used for FQT and data and data generators) has operational fidelity and will correctly represent the operational target system and environment.
- TRL 9 Criterion: ensure that there is a high level of confidence that the test environment used for FQT is a high fidelity representation of the operational environment.

Software TRL for the Development Process

Maturity Status Indicator

- Purpose: to determine that the development process used for the software (used for FQT) development meets the DoD best process as specified by the SEI CMMI Level 3 or higher.
- TRL 9 Criterion: ensure that the FQT software has been developed with a process equivalent to the SEI CMMI Level 3 or higher standards and that there are data and artifacts to provide a process audit trail.

Software TRL for the Safety/Security Requirements Maturity Status Indicator

– Purpose: to determine that the technology adequately meets all of the system safety and security requirements.

– TRL 9 Criterion: ensure that all system safety and security requirements are fully and correctly implemented and have been verified at the completion of FQT for the system under consideration.

Software TRL for the Precision/Performance Requirements Maturity Status Indicator

– Purpose: to determine that the technology adequately meets all of the allocated system precision and performance requirements.

– TRL 9 Criterion: ensure that all allocated system precision and performance requirements are fully and correctly implemented and have been verified at the completion of FQT for the system under consideration.

Software TRL for the Development Environment Maturity Status Indicator

- Purpose: to determine that the development environment is validated, reliable, and predictable, and will correctly and consistently produce the proper results.

- TRL 9 Criterion: ensure that sufficient validation of the development environment has been performed and documented by means of: 1) a formal validation process, or 2) use (without change) of the development environment for similar operational software development that was shown to produce correct and reliable results.

Software TRL for the Code Stability Maturity Status Indicator

- Purpose: to demonstrate that the code is stable, and significant changes are not being made to complete FQT.
- TRL 9 Criterion: the code changes for baselined software used for informal requirements testing (code configuration used to hold TRR and complete FQT) meet the following criterion: code changes < 0.5%

Technology Readiness Level vs. Risk Assessment Matrix

Technology Readiness Level	Non-Safety & Non-Security Software Risk Assessment	Safety & Security Software Risk Assessment
TRL-1	Recommend against use for this deployment	Not Acceptable for Operational Use; Recommend against deployment
TRL-2	Recommend against use for this deployment	Not Acceptable for Operational Use; Recommend against deployment
TRL-3	Recommend against use for this deployment	Not Acceptable for Operational Use; Recommend against deployment
TRL-4	Recommend against use for this deployment	Not Acceptable for Operational Use; Recommend against deployment
TRL-5	High Risk	Not Acceptable for Operational Use; Recommend against deployment
TRL-6	Medium to High Risk	High Risk: Not Recommended for Deployment
TRL-7	Low to Medium Risk	Medium Risk: Not Recommended for Deployment
TRL-8	Low Risk	Low Risk: Advise of the Risk and Recommend Further Technology Maturity Development
TRL-9	No Risk	Low Risk: Acceptable for Operational Use

Software Technology Readiness Assessment Procedure

1. Perform Technology Readiness Assessment of software technologies predicted at project startup.

2. Correlate software TRL assessment with *TRL versus Risk Assessment Matrix*.

Software Technology Readiness Assessment Procedure

3. Include Software Safety Risk Assessment in Risk Mgt. Plan.

4. During system design identify TRLs & Software Safety Risk Assessment for all software.

Software Technology Readiness Assessment Procedure

5. Document the TRL for each software component:
   - Project Plan
   - Software Development Plan
   - Risk Management Plan
6. Make design decision and resource trade-offs based on:
   - S/W TRL Assessment
   - S/W TRLs vs. Risk Assessment
   - Risk Management procedures to optimize for system development and mission success.

Software Technology Readiness Assessment Procedure

7. For TRL's below 8, a Technology Transition Plan is required to reach TRL 8 & 9 at software release/deployment.
    - Schedule
    - Technical Approach
    - Resources
    - Include Technology Transition Planning in the Risk Management Plan.

Software Technology Readiness Assessment Procedure

8. Update the Technology Assessment (TRLs) and Risk Assessment on a continuous basis.
   - Decision Gates
   - Each Major Review (e.g., SRR, SSR, PDR, TRR)
   - Risk Assessment Reviews

9. At time of **System/Software Release,** or **Flight Release,** make final TRL assessment and recommendation including data from the TRL vs. Risk Assessment Matrix, and include in the supportable statement, suitability statement, material release, VDD, and/or other documents as appropriate.

Questions

256-876-9945

willie.fitzpatrick@us.army.mil