

Status of Ongoing Work in Software TRAs/TRLs

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Michael S. Bandor &
Suzanne M. Garcia-Miller

Software & Systems Technology
Conference (SSTC)
29 April 2010



Copyright Notice

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.



Software Engineering Institute

Department of Defense **R&D Laboratory Federally Funded Research & Development Center (FFRDC)**, created in 1984

Under contract to Carnegie Mellon University

Headquarters in Pittsburgh, PA; Offices - Arlington, VA

Stakeholders

- Significant US government sponsors
 - Department of Defense (DoD)
 - Department of Homeland Security (DHS)
- Researchers, developers, users, and acquirers—government, commercial, and academic
- Key industries and organizations with the potential to advance software engineering and related disciplines



Mission and Organization

Mission

The SEI advances software and related disciplines to ensure the development and operation of systems with predictable and improved cost, schedule, and quality.

Technical Programs

Research Programs

- **Software Engineering Process Management (SEPM)**
- **Research, Technology, and Systems Solutions (RTSS)**
- **Computer Emergency Response Team (CERT)**

Application and Support

- **Acquisition Support Program (ASP)**



Background₁

Participated as a TRA Independent Review Team/Panel member since 2007

Assessments across multiple domains; some assessments were software only:

- B-2 Radar Modernization
- C-130 Avionics Modernization Program (AMP)
- Space Command & Control System Increment 1
- Air Force Mission Planning System (MPS)
- Air Operations Center Weapon System (AOC WS)
- Global Positioning System Ground Segment (GPS OCX)
- AOC WS Increment 10.2
- Expeditionary Combat Support System (ECSS)
- Integrated Space Planning and Analysis Network (ISPAN) Increment 2
- Three Dimensional Expeditionary Long Range Radar (3DELRR)



Background₂

Participated as a team member in the TD-1-12 Efforts:

- *“Under the Air Force Smart Operations – 21, Developing & Sustaining Warfighting Systems Core Process” activities, the Technology Development team issued a report (TD-1-12) in Apr 09 with recommendations on how to implement TRAs for software*



Outline

Part 1 – Presented by Mike Bandor

- Issues currently being encountered when assessing specific aspects of software Technology Readiness Levels (TRLs) as they apply to candidates for the Critical Technology Elements (CTEs) list

Part 2 – Presented by Suz Garcia-Miller

- Background and overview of findings and recommendations from the Technology Development 1-12 Software Subgroup (TD-1-12) which reviewed the current TRA process and submitted recommendations for changing the process



Part 1 – Issues Encountered during Technology Readiness Assessments

Mike Bandor



Issues Encountered during TRAs

- Technology choices: high visibility or “glamorous” vs. mundane
- Lack of distinction between software types
- Demonstrations in a “relevant” environment
- TRA process inconsistencies with DoDI 5000.02 acquisition lifecycle
- Definition of a new software technology
- Incomplete consideration of lifecycle maintenance/support
- External influences on technology choices causing an implied CTE
- Technologies started in one increment and finished in a later increment



Technology Choices: High Visibility or “Glamorous” vs. Mundane

Programs acquiring systems with highly visible or “glamorous” technologies may inadvertently overlook more mundane, but nonetheless high impact technologies

- The initial list Technology Elements (TEs) and candidate list of Critical Technology Elements (CTEs) are provided by the Program Management Office (PMO)
- Easy to miss considering important TEs/CTEs unless Independent Review Team (IRT) is fortunate enough to have a member with reasonably intimate knowledge of the program
- IRT & PMO could easily become too focused on the more advanced technology TEs/CTEs presenting a much higher risk to the program
- Software supporting those advanced technologies not considered
 - Size could easily range in size in the multi-millions of SLOC but has yet to be developed and tested!



Lack of Distinction between Software Types

The TRA process does not appear to differentiate between newly developed code, re-used code and COTS software products when applying Technology Readiness Level (TRL) definitions.

- Existing TRA process to assess the “maturity” of the software technology mirrors the hardware technology
- Existing TRA descriptions for the measures of the SW TRLs seem more appropriate to pre-existing software (e.g. COTS S/W)
 - Could be difficult to apply where large amounts of code is a mixture of new or re-used code
 - The meaning of “technical readiness” has different implications for new code vs. COTS S/W – assessing both types with the same definition is difficult
 - Strict interpretation of S/W TRL 5 or higher would exclude all newly developed code that is created post Milestone-B (okay for H/W but not for S/W?)



Demonstrations in a “Relevant” Environment

The software TRL definition, “demonstration in a relevant environment” does not take into consideration who or which “team” performed the prior demonstrations

- Historical trends show prior team integration experience with specific software technologies significantly contributes to reduced programmatic software risk
- Current software TRL definitions appear focused on prior use in similar and relevant environments almost as a “point solution”
- Previous software usage patterns, although important, are also dependent upon the integration experience of the prior development teams
 - Potential significant difference in “technology readiness” with an experienced integration team vs. a completely new integration team
 - Largely an issue for COTS or reused software
 - “Architecting” across the software lifecycle will most certainly not be done exactly the same way by two different teams using the same software technologies
- Technologies demonstrated in a commercial environment do not necessarily map to the anticipated use in military environment



TRA Process Inconsistencies with DoDI 5000.02 Acquisition Lifecycle

The software TRA process appears, in part, to be inconsistent when aligned with certain DoDI 5000.02 program lifecycle model events.

- Current process presents a “chicken-or-the-egg” situation for software TRLs dealing with newly developed code
 - Milestone B (MS-B) requires each CTE be assessed at TRL 6
 - Requires demonstration in a “relevant environment” implying some form of the software architecture exists within that environment to support a demo
 - TRL definitions for software also imply the architecture exists earlier at TRL 5
 - However, a formal S/W architecture doesn’t exist until the Preliminary Design Review (PDR), which occurs after MS-B for non-Major Defense Acquisition Programs (MDAPs)* - too late for use in a MS-B TRA
 - The PDR may be too high-level of design to ascertain the architecture for the “relevant environment”
 - PMOs sometimes claim the intent to “reuse” important attributes or components of whatever demo architecture existed pre-MS-B, but no enforcement mechanism exists to ensure compliance with the reuse

**Prior to the Dec 2008 update of DoDI 5000.02, PDRs were after MS-B*



Definition of a New Software Technology

The TRL “threshold” for defining the definition of a software technology maturity seems vague enough that “how to consistently apply it” to new, reused, and COTS software technologies is subject to interpretation.

- Large software technologies are often bundled, requiring decomposition into specific parts
 - Process should be accomplished by the Program Management Office (PMO) not the TRA team
- Technology Elements should be defined and then potential CTEs identified
 - The TRL level guidance should be used in the determination
 - Tendency by engineers to use the TRLs as a justification why something ISN'T a CTE (e.g., “it’s not a CTE, it’s an engineering/integration issue”)
- Software technologies that separately would rate a high TRL, combined with other technologies for the first time should cause another look at the TRL with respect to the integration issues
- Embedded software can be problematic
 - It could be rated at a lower TRL than the hardware it runs on & vice versa



Incomplete Consideration of Lifecycle Maintenance/Support

There is not enough consideration in the TRA process directed toward the lifecycle maintenance/support of technologies.

- Largely due to changes/updates being driven by corporate market dynamics
- Changes not under control or under the influence of the PMO!
 - On programs with long-development time, a chosen COTS software product may become obsolete before the initial system is fielded
 - Consider the requirement to require PMOs to provide COTS upgrade and technology refresh plans and activities as part of the materials provided to the IRT
- Risk management of TEs and CTEs
 - No mechanism exists for reporting non-technology related risks/concerns identified by the IRT during the TRA process
 - All candidate CTEs that didn't formally qualify as CTEs were considered for a specific reason
 - Consider placing on a "watch item" list to manage the risk potential



External Influences on Technology Choices Causing an Implied CTE

As more programs are being hosted by other organizations, technology choices or upgrades to those environments (not directed or caused by the program) may cause an “implied” CTE.

For example:

- Host changes to a virtual server environment
- Program was designed to run in a more traditional N-tier environment depending upon physical servers being deployed
- It affects the cost & schedule of the program (cost savings from not having to buy & deploy additional physical servers)
- It causes a “new relevant environment” to be realized
- By the TRA guidelines, the server virtualization would be considered a CTE

However, a problem now arises

- The program does not have a virtual server requirement, nor a performance requirement that the CTE can be traced to (part of the TRA process)



Technologies Starting in One Increment & Finished in a Later Increment

Programs executing an incremental acquisition strategy may chose to initially implement a technology in one increment, however the full implementation of the functions are done in a subsequent increment (full capabilities)

- Potential exists for a TE/CTE to be missed as a result
- Does the technology get evaluated as a partial fulfillment of the function or as a full implementation later, or both?
 - Potential exists to have its status change between increments, affecting programmatic decisions by the PMO
- Not really addressed in the current guidance



Part 1 - Summary

The issues in evaluating software CTEs are still problematic

- There isn't a single answer that covers every domain and every situation
- Technology choices greatly affect candidate CTE evaluations
- Current guidance doesn't distinguish between new code, reused code, and COTS/GOTS software
- More up-front engineering and architecture work is required for software in order to meet the intent of Milestone-B
- Caution is needed when using the term "new" technology – new to what & whom? Is it a "critical" technology or an integration/engineering problem?
- Long-term support needs to be considered (technology refresh, upgrades, obsolescence, etc.)
- Be aware of external influences (hosting environments, etc.) that may cause a program to rethink the architecture and "relevant" environment choices
- PMOs need to give some thought to the technology implementation relative to the incremental development strategy; don't lose sight of a potential CTE between increments



Part 2 – Background & Overview of Findings & Recommendations from the Technology Development 1-12 Software Subgroup (TD-1-12)

Suz Garcia-Miller



The Team: AFSO-21 D&SWS TD-1-12

Summary of Software Technology Readiness Assessment Recommendations

December 16, 2008

Prepared by the TD-1-12 Software Sub-team:

Thomas Christian (AFMC/ASC),

Suzanne Garcia (SEI),

Peter Hantos, Team Leader (The Aerospace Corporation)

William Nolte (AFMC/AFRL),

Paul Phister (AFMC/AFRL)



Software Engineering Institute

Carnegie Mellon

Status of Ongoing Work in Software
TRAs/TRLs - SSTC 2010
Bandor & Garcia-Miller, 29 Apr 10
© 2010 Carnegie Mellon University

20

Task Context-Constraints

Constraint #1: Due to DoD-SAF/AQ direction, the Software Sub-team was not permitted to change the basic definitions of Technology Readiness Levels.

Constraint #2: Also due to DoD-SAF/AQ direction, a uniform, milestone-independent rating scheme is expected for TRL determination.

Constraint #3: Software TRA recommendations cannot represent an explicit or implied request for changing the acquisition process as it is currently outlined in DoD 5000 or its National Security Space equivalent, NSSAP 03-01.

Constraint #4: Due to Public Law, the requirement is to evaluate and certify CTEs in a Relevant Environment at Milestone B of the DoD 5000 (or KDP-B of the NSSAP 03-01) acquisition process. Note that the law itself does not explicitly mention TRLs.



Team Recommendations

- Recommendation 1:** Introduce a clear definition of software technology as a foundation for the development of evaluation criteria and process.
- Recommendation 2:** Introduce an unambiguous approach to deal with algorithms.
- Recommendation 3:** Emphasize the role of Software Architecture in mining software technology-related information during software TRAs.
- Recommendation 4:** Use a customized, TRA-oriented version of the architectural viewpoints specified by the IEEE architecture standard.
- Recommendation 5:** Introduce a clear definition of Relevant Environment for Space.
- Recommendation 6:** Clarify TRL goals and knowledge required for their assessment.
- Recommendation 7:** Use a framework-based approach for TRL determination
- Recommendation 8:** Include explicit references to Network Centric Warfare (NCW) and Service Oriented Architecture (SOA) in the CTE identification guidelines.
- Recommendation 9:** Keep and analyze a running log of CTEs.



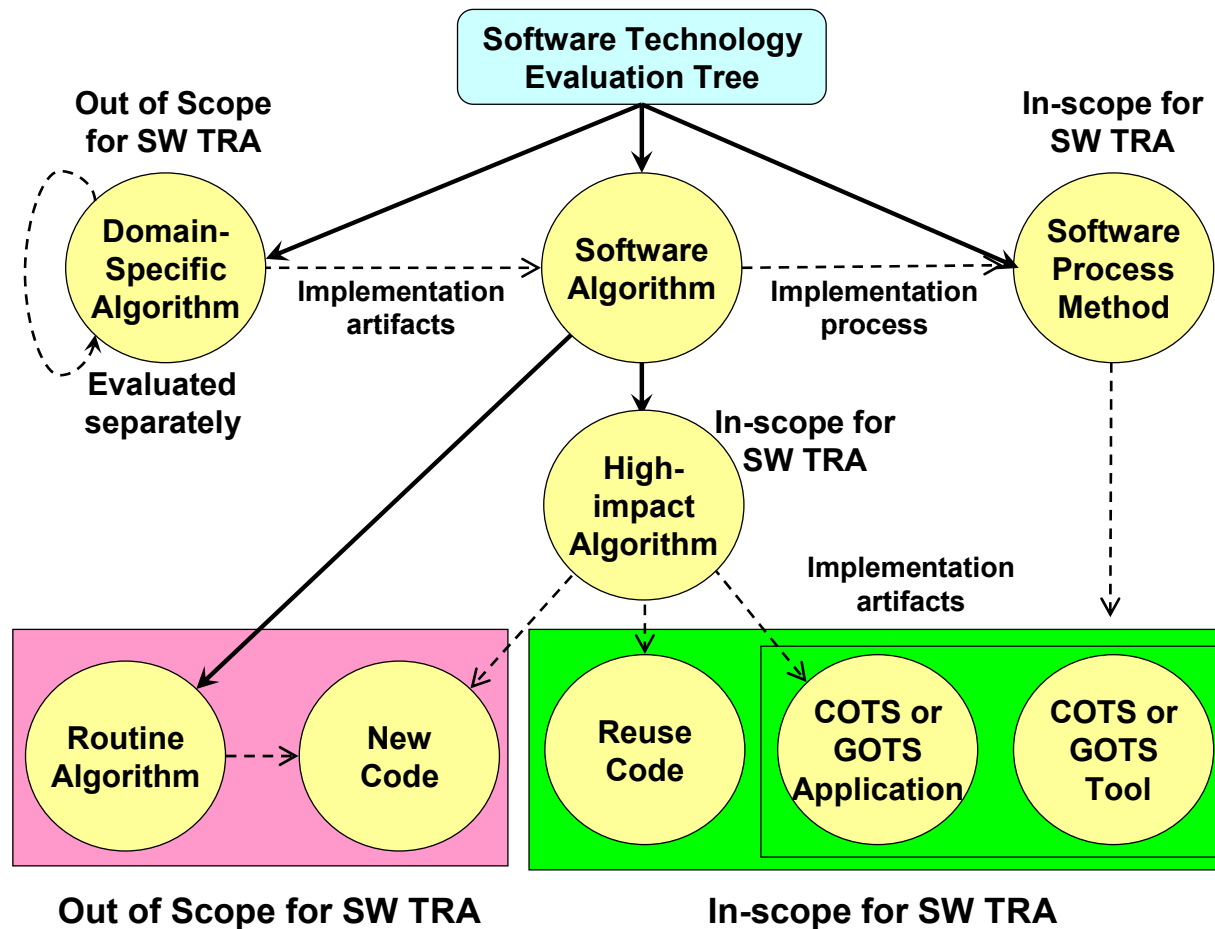
How the Team Views SW Tech Readiness

Software Technology Readiness is a state of understanding from which the software can be designed and implemented with predictable performance, costs, delivery, and quality characteristics.

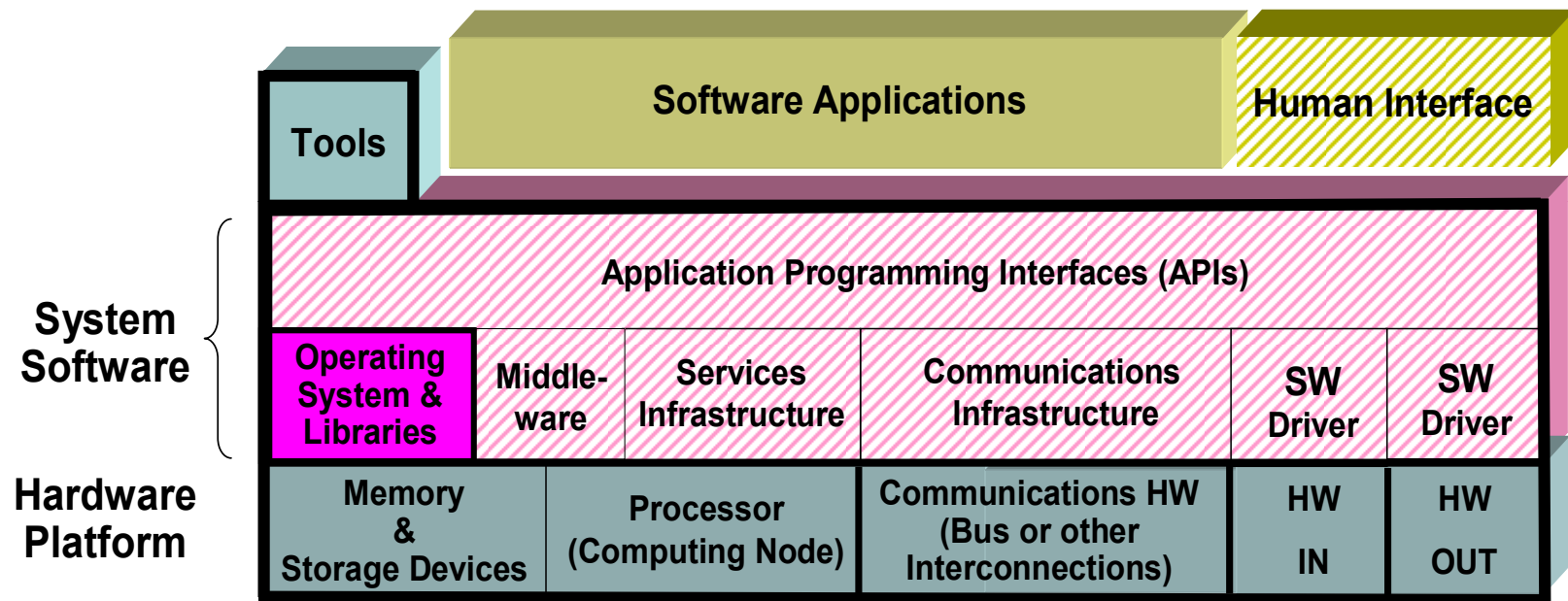
TRL \geq 6 software technology maturity ratings indicate a high level of confidence in that no special solutions would have to be invented beyond normal software engineering practices to satisfy the planned mission requirements.



How SW CTEs with High Algorithmic Content Should Be Selected



Internal, Physical Environment for Software



SW Tech Readiness Determination Framework

Idea₁

| # | Basic TRL DEFINITIONS from TRA Deskbook | TRL GOALS | Knowledge Involved in Achieving HW Objectives | Knowledge Involved in Achieving SW Objectives |
|---|---|-------------------------------------|---|---|
| 1 | Basic principles observed and reported | Demonstrate scientific feasibility | Natural Sciences | Computer Science |
| 2 | Technology concept and/or application formulated | | Natural Sciences | Computer Science |
| 3 | Analytical and experimental critical function and/or characteristic proof of concept. | | Natural Sciences | Computer Science |
| 4 | Component and/or breadboard validation in a laboratory environment | Demonstrate engineering feasibility | Hardware Engineering | Software Engineering |
| 5 | Component and/or breadboard validation in a laboratory environment | | Hardware Engineering | Software Engineering |
| 6 | System/subsystem model or prototype demonstration in a relevant environment | | Hardware Engineering | Software Engineering |
| 7 | System prototype demonstration in an operational environment | Demonstrate operational feasibility | Systems Engineering, Hardware Engineering | Systems Engineering, Software Engineering |
| 8 | Actual system completed and qualified through test and demonstration | | Systems Engineering, Hardware Engineering | Systems Engineering, Software Engineering |
| 9 | Actual system proven through successful mission operations | Demonstrate operation | Domain | Domain |



Software TRL Eval Dimensions – Information Need for Goal Satisfaction Validation₁

Artifacts

- During the discussion of algorithms it was explained how the algorithms go through a certain metamorphosis from ideas to implementation during technology maturation. The purpose of this evaluation dimension is to identify and track the stages of this change.

Structural Context

- As it was stated earlier, the integration activity itself is not in scope for TRA. However, potential interference from other CTE's or simply other parts of the system is a valid concern, and to address these potential problems the structural context need to be gradually expanded during TRL demonstration.

Software Environment

- The focus of this inquiry is the elements of the environment that are needed for developing and operating the objective system and they are appropriate for the validation of the TRL goal satisfaction. This includes considerations for hardware/software interfaces and miscellaneous other, technology inter-dependencies as well.



Software TRL Eval Dimensions – Information Need for Goal Satisfaction Validation₂

Validation Environment and Methods

- The focus of this inquiry is specifically on the elements of the software environment that are needed for validating the TRL goal satisfaction.
- Data Used for Validation
- The focus of this inquiry is to verify that the appropriate data has been used for validating the TRL goal satisfaction.

Configuration Management (CM)

- A critical aspect of technology maturity is repeatability. In the TRL 4-6 range software prototypes are used to demonstrate engineering feasibility. Applying CM in both experimental and operational-like environments is a prudent and necessary practice to enable accurate validation of the TRL goal satisfaction. In case of $TRL \geq 7$ productization of the technology has already been started, consequently Configuration Management needs to be carried out in compliance with production environment rules.

Documentation

- Documentation's role in technology maturity demonstrations is similar to CMs. The provided documentation needs to be both appropriate and sufficient to facilitate the validation of the technology's expected performance.



Demonstrating Scientific Feasibility – TRL 1-3

| TRL | Basic TRL DEFINITIONS | Artifacts | Structural Context | SW Environment | Validation Environment and Methods | Data Used for Validation | Configuration Management | Documentation |
|-----|---|---|--------------------|--------------------------|---|-------------------------------|--------------------------|---|
| 1 | Basic principles observed and reported | Research articles, peer-reviewed white papers, point papers, early conceptual models | n/a | “Academic”, experimental | Basic research using analytical Methods | n/a | n/a | Appropriate and sufficient to demonstrate basic principles |
| 2 | Technology concept and/or application formulated | Analytic studies, papers on competing technologies | n/a | “Academic”, experimental | Applied research using analytical Methods | Synthetic data only | n/a | Appropriate and sufficient to demonstrate application concept |
| 3 | Analytical and experimental critical function and/or characteristic proof of concept. | Analytical and simulation models; Availability of appropriate COTS/GOTS or reusable software artifacts is explored. | n/a | “Academic”, experimental | Active R&D initiated via the use of models and simulation | Partially representative data | n/a | Appropriate and sufficient to interpret analytical or experimental data; Full documentation is available on COTS/GOTS or reusable software under consideration. |



Demonstrating Engineering Feasibility – TRL 4-6

| TRL | Basic TRL DEFINITIONS | Artifacts | Structural Context | SW Environment | Validation Environment and Methods | Data Used for Validation | Configuration Management | Documentation |
|-----|---|---|------------------------|--------------------------|--|---|--|---|
| 4 | Component and/or breadboard validation in a laboratory environment | A stand-alone prototype solving a partial-scale problem; Proposals for a nominal software architecture and for a simulation/stimulation work-up plan; COTS/GOTS, or reusable SW if applicable | Prototype SW Component | “Academic”, experimental | Advanced technology development with throwaway or evolutionary SW prototypes | Representative data | Limited scope; appropriate for experimental environment | Appropriate and sufficient to interpret experimental results; Full documentation on chosen COTS/GOTS or reusable software |
| 5 | Component and/or breadboard validation in a laboratory environment | A stand-alone prototype solving a partial-scale problem; Detailed software architecture and final simulation/stimulation work-up plan; COTS/GOTS, or reusable SW if applicable | Prototype SW Component | Operational-like | Advanced technology development with throwaway or evolutionary SW prototypes | Representative data | Appropriate for operational-like, production environment | Appropriate and sufficient to interpret results; Full documentation on chosen COTS/GOTS or reusable software |
| 6 | System/subsystem model or prototype demonstration in a relevant environment | Viable prototype providing the foundation for productization | Prototype WBS Level 3 | Operational-like | Development using evolutionary SW Prototype | High-fidelity data representative of relevant environment | Appropriate for operational-like, production environment | Appropriate and sufficient to validate relevant environment and interpret demonstration results |



Demonstrating Operational Feasibility – TRL 7-8

| TRL | Basic TRL DEFINITIONS | Artifacts | Structural Context | SW Environment | Validation Environment and Methods | Data Used for Validation | Configuration Management | Documentation |
|-----|--|-----------------------|--------------------|-----------------------|--|---|--|---|
| 7 | System prototype demonstration in an operational environment | Productized component | WBS Level 2 | Actual SW Environment | End-to-end testing of Production SW using system simulator | High-fidelity data representative of relevant environment | Appropriate for the actual environment | Appropriate and sufficient to validate test results |
| 8 | Actual system completed and qualified through test and demonstration | Productized component | WBS Level 1 | Actual SW Environment | Testing of Production SW using OT&E | Real data | Appropriate for the actual environment | Appropriate and sufficient to validate technology's performance and operate and maintain the product. |

Demonstrating Operation – TRL 9

| TRL | Basic TRL DEFINITIONS | Artifacts | Structural Context | SW Environment | Validation Environment and Methods | Data Used for Validation | Configuration Management | Documentation |
|-----|--|-----------------------|--------------------|-----------------------|------------------------------------|--------------------------|--|---|
| 9 | Actual system proven through successful mission operations | Productized component | WBS Level 1 | Actual SW Environment | Actual | Real data | Consistent with the actual environment | Appropriate and sufficient to validate technology's performance and operate and maintain the product. |



Dependency Considerations for Elements of the Software Environment

Due to the dependency of software on its environment of operation, we can establish the basic rule that the TRL rating for any software element cannot be higher than the TRL rating of the lower layers.

This rule also comprehends the software's dependency on the hosting hardware's technology maturity. In mathematical terms, the following relationships apply:

$$\text{TRL}(\text{SW Applications \& Human Interface}) \leq \text{TRL}(\text{Tools \& System SW}) \leq \text{TRL}(\text{Hardware Platform})$$



Technology Maturity vs. Product Maturity

A closer analysis of TRL goals and evaluation dimensions shows that while the TRA process' stated objective is to track **technology maturity**, at TRL 6-9 actually we are tracking selected aspects of **product maturity**.

- This ambiguity is the basis for substantial critique of the TRA process, claiming that it does not really add value and conventional program management risk reduction efforts that are already in place should suffice.
- The lack of clear definition of technology further exacerbated this problem and particularly the definition of software technologies proved a very contentious issue.

TRLs as Snapshots

The current process treats TRLs as snapshots with the following objectives:

- TRL 1-3: Demonstration of Scientific Feasibility
- TRL 4-6: Demonstration of Engineering Feasibility
- TRL 7-8: Demonstration of Operational Feasibility
- TRL 9: Demonstration of Mission Operations



Asymmetrical Relationship between Systems Engineering and Software Engineering

This concern is related to the milestone-independent nature of the current TRA process, i.e., that the used, 9-level metrics are the same at every milestone, for every involved discipline, such as systems engineering, hardware engineering, and software engineering.

However, there is an asymmetrical relationship between Systems Engineering and Software Engineering Processes.

- As a result, at any given time the amount, quality, and depth of software engineering information is not on par with the available systems engineering information.
- This discrepancy is also fueling the push-back on software CTEs, particularly at early milestones, because in most cases there is not enough software information available to answer the TRA Deskbook's questions.

Unfortunately, the mentioned relationship and its consequences are not widely understood by most of the stakeholders.



Part 2 - Summary

The AFSSO-21 Software Subteam covered a fairly wide set of issues related to SW TRAs/TRLs

- Many of those issues parallel ones brought up elsewhere in today's presentations

Constraining the team to work within the existing DoD TRL framework prevented us from going too far out into new territory; however

- It forced us to analyze in greater depth what are the assumptions underneath TRLs so we could try to find the points of synergy that do exist

Although we acknowledge the need and desire for a single index number as a way of communicating the summary of findings in a TRA

- We believe that a profiling approach that provides a richer characterization is essential to ensure that TRA and engineering teams can effectively communicate technology and product maturity issues
- Although we developed the profiling approach for software, our instinct is that it would have utility for other types of CTEs as well



Contact Information

Michael S. Bandor

*Senior Member of the Technical Staff
Acquisition Support Program (ASP)*

Telephone: +1 210-380-5563

Email: mbandor@sei.cmu.edu

Suzanne M. Garcia-Miller

*Senior Member of the Technical Staff
Research Technology & System Solutions
(RTSS)*

Telephone: +1 412-268-9143

Email: smg@sei.cmu.edu

World Wide Web:

www.sei.cmu.edu

www.sei.cmu.edu/contact.html

U.S. mail:

Software Engineering Institute

Customer Relations

4500 Fifth Avenue

Pittsburgh, PA 15213-2612

USA

Customer Relations

Email: customer-relations@sei.cmu.edu

Telephone: +1 412-268-5800

SEI Phone: +1 412-268-5800

SEI Fax: +1 412-268-6257

