Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

An Alternative to TRLs for COTS Software-Intensive Systems

Jim Smith

Sponsored by the U.S. Department of Defense

Purpose

To introduce an alternative to Technology Readiness Levels (TRLs)—**ImpACT**—as a way to reason about the suitability of Commercial-Off-The-Shelf (COTS) software products within a given context.

Overview

Background: TRLs

Multiple Aspects of Readiness
- Quality and readiness
- Time-varying effects

The "Problem" with TRLs

Introduction to ImpACT

Comparison of TRLs and ImpACT

Background: $TRLs_1$

TRLs initially developed by the National Aeronautics and Space Administration (NASA) to aid in assessing technology maturity risk.

- Used routinely since early 1990's

TRLs provide a scale to gauge the readiness of a product or technology for use within a particular context. For example, TRL 1 (the lowest level) is defined as:

> *Lowest level of technology readiness. Scientific research begins to be translated into applied research and development. Examples might include paper studies of a technology's basic properties.*

Background: $TRLs_2$

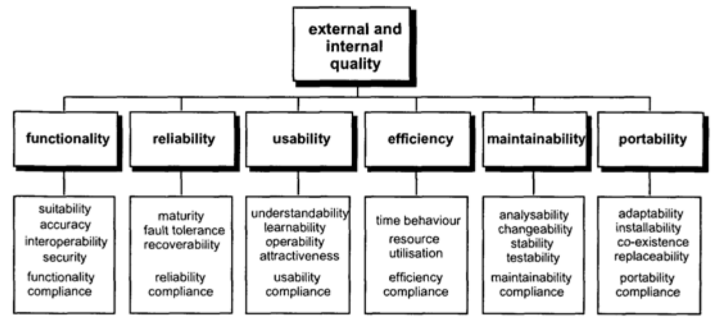
There is growing acceptance within the U.S. Department of Defense (DoD) for the use of TRLs.

- Current guidance requires programs perform a technology readiness assessment prior to System Development and Demonstration (SD&D)
- Recommended minimum maturity for a technology to be included in an acquisition is TRL 6 (system/ subsystem model or prototype demonstration in a relevant environment)
  - TRL 7 (system prototype demonstration in an operational environment) is preferred

Use of TRLs *strongly encouraged* by the General Accounting Office (GAO).

Relationship Between Quality and Readiness

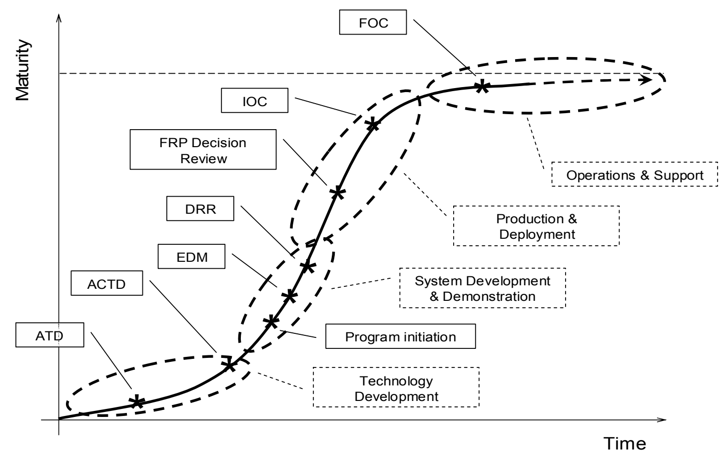
ISO/IEC-9126 defines software quality models which include internal and external quality attributes, as well as quality in use factors.



"Readiness" can be viewed as characteristic of a combination of quality attributes *in the context of a particular system development or acquisition*.

Time-Varying Influences on Readiness

Hanakawa, *et al*, have developed a model of how knowledge is gained during software development.* By analogy, this provides some insight into the tolerance for various sources of risk at any point in the development.



Thus, at a given time, different contributors to risk may be more or less important than at other times.

---

*Hanakawa, N.; Morisaki, S.; & Matsumoto, K. "A Learning Curve Based Simulation Model for Software Development," 350-359. *Proceedings of the 20th International Conference on Software Engineering*. Kyoto, Japan, Apr. 19-25, 1998. New York: IEEE Computer Society Press, 1998.

The “Problem” with TRLs$_1$

TRLs were not initially developed for software. There have been some attempts to apply them to both *software* and *systems*:

- U.S. Army Communications Electronics Command (CECOM), in collaboration with the Software Engineering Institute (SEI), has developed draft software TRLs to improve their maturity assessments for new information assurance technologies
- U.S. Air Force Research Laboratory (AFRL) has developed a TRL calculator, using the NASA definitions for hardware and the CECOM software definitions

The “Problem” with TRLs$_2$

The difficulties in using TRLs for software—especially COTS software—seem to take four principle forms:

- TRLs blur together all aspects of readiness, making it difficult (or impossible) to understand the contributions of any particular element of readiness to the overall product readiness
- TRLs do not directly account for the criticality of a product or technology to the system (or the difficulty of replacing it)
- TRLs ignore the effects of software “decay” or COTS product obsolescence
- Since TRLs blur together the different elements of readiness, they can’t accommodate the notion of different elements varying in their relative importance over time

An Alternative: ImpACT

Since “readiness” is really a time-varying function of multiple attributes in a given context, it makes sense to have a multi-attribute readiness indicator and reasoning approach which directly addresses these issues. ImpACT (short for **Imp**ortance—**A**vailability—**C**riticality—**T**imeframe) represents an initial attempt at this.

Important Safety Note: ImpACT describes a general methodology and framework for reasoning about COTS software product readiness. ImpACT does not dictate any particular evaluation methodology or process… Selection of a suitable evaluation framework must be informed by the system development context.

Definition of ImpACT Attributes

Readiness is contextually-driven, and is a function of "some number" of factors. ImpACT comprises four factors, which are assessed with regard to level of risk:

- **Imp**ortance – the criticality of the software technology or product to the system (or how closely-tied is the system to a specific product or technology)
- **A**vailability – the "COTS-ness" of a product or technology
- **C**apability – a measure of the functional fit between the product or technology and the system's requirements
- **T**imeframe – an indication of how well the product or technology lifecycle coincides with the needs of the system

ImpACT - Importance

The **Imp**ortance factor is assigned a value from 0 to 5 (lowest to highest risk) on the following scale:

0: At least one suitable alternative can be “plugged-in” with minimal tailoring

1: At least one suitable alternative available; reintegration and minimal software changes required

2: Moderate reintegration and pervasive software changes required

3: Significant architectural and/or implementation changes required in part of the system

4: Significant, pervasive architectural and/or implementation changes required

5: No suitable work-around… back to the drawing board!

ImpACT - Availability

Availability is assigned a value from 0 to 5 as follows:

0: Widespread commercial use; “shrink-wrap”
1: Limited commercial use; first commercial use in particular domain
2: Public testing (public beta, release candidate)
3: Internal testing (alpha/beta testing)
4: “Opportunity ware”
5: “Vapor ware”

ImpACT - Capability

Capability is assigned a value on the following scale:

0: “Perfect” fit between product capabilities and system requirements

1: Desired functionality present, but some minor “fit” issues

2: Deficiencies in one or more second- or third-tier functionality; work-arounds available

3: Deficiencies in second- or third-tier functionality with no work-arounds

4: Significant deficiencies in one or more critical functional areas; degraded performance with work-arounds

5: Critical functions missing; no work-arounds. Unsuitable.

ImpACT - Timeframe

Timeframe is assigned a value from 0 to 5 on the following scale:

0: Product or technology available over the projected life of the system

1: Available when needed; retirement/replacement anticipated during system life

2: Available when needed; end-of-life with replacement announced

3: Available when needed; end-of-life without known replacement announced

4: No longer available by system “need date,” but replacement announced or alternate product available

5: No longer available by system “need date,” and no replacement or alternate product available

Sample Application$_1$

The following slides will demonstrate using ImpACT for a trivial case of evaluating two products (“A” and “B”) for use in a system under consideration, and compare this with the results obtained by using TRLs.

Sample Application$_2$

First, what would TRLs tell you about the readiness of these two products?

In this example, products A and B have been developed for use in the same “domain.” Product “A” is in pre-release (public beta testing); product “B” is currently available as a “shrink-wrapped” product. On the TRL scale, product “A” would be roughly equivalent to level 7-8; similarly, product “B” would be evaluated at level 9.

Sample Application$_3$

So… How would you use ImpACT in this example?

First, a determination of the relative importance of the readiness criteria is required. Where is the system in its life cycle? How important is it for a product to be highly stable? How much coupling between the system and the product is acceptable? For this example, assume:

- Importance (I) is determined to be somewhat more important than Capability (C)
- C, in turn, is judged to be much more important than Availability (A)
- A is considered to be more important than Timeframe (T)

In other words:

$$I > C >> A > T$$

Sample Application$_4$

Second, an evaluation framework must be defined. In this case, Saaty's Analytic Hierarchy Process (AHP) was used. Thus, a pairwise comparison matrix (PCM) of the readiness criteria can be expressed as:

$$
\begin{array}{c} \\ I \\ A \\ C \\ T \end{array}
\begin{array}{c} \begin{array}{cccc} I & A & C & T \end{array} \\ \begin{bmatrix} 1 & 7 & 2 & 9 \\ 1/7 & 1 & 1/3 & 1 \\ 1/2 & 3 & 1 & 7 \\ 1/9 & 1 & 1/7 & 1 \end{bmatrix} \end{array}
$$

Sample Application$_5$

The products are next evaluated against the ImpACT criteria, with respect to the system under consideration, the degree of risk aversion/tolerance of the developer or acquiring organization, etc. (in other words, the context for the development).

In this example, Product "A"

- Supports loose coupling with the system (i.e., it can be "easily" replaced) → I=0
- Is currently undergoing public beta testing → A=2
- Lacks some non-critical functions, but equivalent functionality can be obtained through operational work-arounds → C=2
- Is currently available, but is anticipated to be replaced during the system's lifetime → T=1

Sample Application$_6$

A similar analysis is performed for product "B," and the resulting ImpACT values for both products can be represented as:

$A_{IACT} = [0,2,2,1]$ $B_{IACT} = [2,0,3,2]$

For this context, it was determined that a suitable mapping from ImpACT criteria to AHP weightings was:

ImpACT	0 =	AHP	1 (roughly equal)
	1 =		2 (slightly more important than)
	2 =		3 (more important than)

Thus, the PCM for products "A" and "B" with respect to the Importance criterion is:

$$\begin{array}{c|cc} & A & B \\ \hline A & 1 & 3 \\ B & 1/3 & 1 \end{array}$$

Sample Application$_7$

By a similar process, the PCMs for the remaining criteria can be determined, and applying the AHP method* results in weighted scores for the two products as shown:

Product "A" = 0.68
Product "B" = 0.32

Therefore, within the context of this *extremely* trivial example, product "A" is evaluated significantly higher than product "B" with the ImpACT criteria—which is the exact opposite of the results obtained by using TRLs.

---

*Which is left as an exercise for the audience members…☺

Summary/Conclusions

While TRLs have proven themselves useful in assessing the relative maturity of competing products and technologies, there are several challenges in using them to assess software products and technologies—especially COTS software products.

ImpACT represents an early attempt at addressing the shortcomings of TRLs in this area. This approach needs to be validated (refined? extended?) through application in actual practice.

Bottom line: ImpACT appears to provide greater insights into the readiness of a software product or technology than that obtained with TRLs, but ImpACT is still only *part* of an overall risk assessment.

Questions…?

Contact Information

Jim Smith
Software Engineering Institute
4301 Wilson Boulevard, Suite 902
Arlington, VA 22203-4191
(703) 908-8221
jds@sei.cmu.edu
http://www.sei.cmu.edu/staff/jds/