ImpACT: An Alternative to Technology Readiness Levels for Commercial-Off-The-Shelf (COTS) Software

James D. Smith II

Carnegie Mellon Software Engineering Institute
4301 Wilson Boulevard, Suite 902, Arlington, VA 22203
jds@sei.cmu.edu

Abstract. The use of Technology Readiness Levels (TRLs) as a tool in assessing acquisition and development program risk has steadily increased over the past several years. There is considerable evidence to support the utility of using TRLs as part of a risk assessment, but there are some difficulties in using TRLs with software, especially Commercial-Off-The-Shelf (COTS) software technology and products. These difficulties take several forms, including "blurring-together" various aspects of COTS technology/product readiness; the absence of some important aspects of readiness; COTS product "decay;" and no mechanism to account for changes in the relative importance of the contributors to technology/product readiness over time. This paper briefly examines these issues, and proposes an alternate methodology—ImpACT—for assessing COTS software technology and product readiness which considers these factors.

1 Introduction

Technology Readiness Levels (TRLs) have been used within the National Aeronautics and Space Administration (NASA), as part of an overall risk assessment process, since the late 1980s. By the early 1990s, TRLs were routinely used within NASA to support technology maturity assessments and comparisons of maturity between different technologies [1, 2].

Within the United States Department of Defense (DoD), there has been considerable interest in using TRLs as part of risk assessments for entire systems, including hardware and software. Current DoD guidance requires technology readiness assessments prior to entering System Development and Demonstration; TRLs are one approach to meeting this requirement [3]. The Air Force Research Lab has adapted the NASA TRLs for use in assessing the readiness of critical technologies for incorporation into weapon systems, and the Army Communications Electronics Command (CECOM) has developed a draft set of TRLs to support software technology management [4, 5].

Several sources cite the difficulties in applying TRLs to assess the readiness of software-based technologies and products [5, 6]. Some of the characteristics of TRLs that affect their use in assessing COTS software technology and product readiness are discussed in more detail in the following sections.

1.1 “Blurring” Together Multiple Aspects of Readiness

One of the difficulties with using TRLs in readiness assessments involving COTS software technology and products is the way in which TRL definitions combine, or “blur” several different aspects of readiness. Because different aspects of maturity may be more-or-less important within a particular program context, this makes it difficult to understand the contributions of individual readiness components to the overall product or technology readiness. Thus, the implications of these individual aspects to the overall readiness of a product or technology—and by extension, to the overall risk assessment—are difficult to discern.

As an example, in their draft TRLs for software, CECOM defines TRL7 as

> *Represents a major step up from TRL6, requiring demonstration of an actual system prototype in an operational environment… Algorithms run on processor of the operational system and are integrated with actual external entities. Software support structure is in place. Software releases are in distinct versions. Frequency and severity of software deficiency reports do not significantly degrade functionality or performance. VV&A complete. [5].*

Note how this definition combine different aspects of technology readiness (e.g., functionality, maintainability, and reliability), making it all but impossible to understand how the contributions of any one facet of readiness affects the overall product or technology maturity.

1.2 Product/Technology Criticality

Just as importantly, TRLs leave out such considerations as the degree to which the technology is critical to the overall success of the system (including how difficult it would be to replace it, or assume some fall-back posture, should the technology in question prove unacceptable), or the suitability of the technology in question to its intended use within the system.

Some programs have attempted to deal with this effect by correction factors to adjust the TRL of a given technology for, say, the criticality of that technology to the success of the system, or the technical complexity of the technology [7].

1.3 COTS Software Product “Decay”

TRLs are used to gauge technology or product readiness growth within a specific context; this information is then used to inform judgments about the maturity of a given product or technology for use in a particular operational environment. Therefore, a product assessed as being at TRL 9—the highest level—is viewed as being at low risk to the system. The fact that a product or technology is subject to “decay,” (e.g., replacement by a newer release, or product of unproven maturity, or simply retired without replacement), is not accounted for in the existing TRL definitions.

1.4 Time-Varying Contributions to Readiness

Another problem with existing TRL definitions is that for different phases of an acquisition, or different lifecycle phases within a development, different technology readiness levels may be appropriate *for different reasons*. For example, in a "proof of concept" demonstration, it is reasonable to use relatively immature, untested technologies as part of the demonstration. On the other hand, for a program in post-deployment sustainment, use of "tried and true" technologies and products is prudent.

Various programs have attempted to deal with this by applying some form of "correction factor" to a raw TRL. One approach used is to adjust a TRL downward by some amount if a particular technology or product comprises more than some percentage of the functionality of the system [7]. Another technique used is to "normalize" technology maturity to the relevant environment for the different lifecycle phases of an acquisition or development (e.g., for a laboratory "benchtop" test, a product or technology with a TRL of 3 or 4 may be acceptable) [4, 5]. Neither of these approaches addresses the issue of how different aspects of maturity contribute—in varying degree—to defining the context in which to understand technology or product maturity during different phases of system acquisition and development.

2 An Alternative Approach

The previous section outlines some of the challenges encountered in using TRLs to assess programmatic and technical risks for systems that include COTS software products. The remainder of this paper will propose an alternative approach that addresses these issues.

2.1 Background

This work grew out of an earlier SEI effort to develop a framework for evaluating COTS technology maturity. A key contribution of this effort was the recognition that technology readiness—especially for COTS software products and technology— is not something that can be represented by a single number. Rather, technology readiness is the result of several factors within a particular context, similar—in concept—to the definition of the Modeling and Simulation Technology Readiness Levels (M&S TRLs) proposed by the Department of Energy, which defined modeling and simulation technology readiness as a function of "correctness," "usability," and "relevance" [8].

In this initial framework, the readiness attributes were:

- When needed (and impact to system if not available)
- Off-the-shelf (verification)
- When projected to be ready (and level of investment needed to bring to that point)
- Projected date of obsolescence (and impact to system)

To use this within a program, software products or technologies would be evaluated against each of the attributes, and assigned values (red/yellow/green) according to the degree to which each product satisfied the conditions described by the attributes. As an example, for the “Off-the-shelf” criterion, a rating of “Green” would mean that the product was in broad commercial use, “Yellow” would indicate that the product had been demonstrated in a relevant environment, and “Red” would signify that your system would represent the first use of the product in question. In addition, a product was rated “High,” “Medium,” or “Low” based on its importance to the system.

2.2 ImpACT: A Multi-Attribute Readiness Description

This initial attribute set afforded greater insight into technology maturity than that obtained from TRLs, but there were difficulties in using it to evaluate COTS technology and product readiness. First, these attributes had the same non-orthogonality problem as TRLs. For example, the “when needed (and impact to system if not available)” attribute combined elements of both time and criticality into a single rating. Similarly, there was a blurring of concerns in each of the initial attributes, resulting in little quantifiable improvement in understanding software technology and product readiness than that provided by TRLs.

This experience led to a realization that a different set of maturity attributes was necessary. From our previous experience with COTS-based systems, and after an extensive (though by no means exhaustive) review of the available literature, four main factors emerged as key elements in addressing the software product/technology maturity “problem”:

1. <u>The criticality of the software technology or product to the system</u>. To what degree is the technology or product essential for the successful operation of the system, as well as the amount of disruption to the system should the product or technology not work, or not be available? This attribute reflects how closely the system architecture and/or implementation is tied to a particular technology or product, and is an indication of the potential for technology or vendor “lock,” where it is extremely difficult—or impossible—to substitute a different technology or product within a system.
2. <u>The availability of the software technology or product</u>. Is this a commercially available product in widespread use in relevant domains? Or, is it a home-grown engineering tool with no commercial support?
3. <u>The degree of “fit” or “misfit” of the software technology or product and the system</u>. Does it have the necessary functionality? If not, how critical is the missing functionality? Are there unneeded capabilities? Do these introduce other complications (e.g., vulnerabilities, undesired modes of operation, etc.)?

4. How well the product or technology lifespan matches the needs of the system. Will a product or technology be ready when needed? Will it still be supported by its developer after incorporation into the system, or will it become an "orphan"?

These factors were then refined into a revised attribute set—ImpACT—defined as:

- ***Imp****ortance* – Criticality to the system; difficulty of effecting a work-around if the technology or product doesn't work (or isn't available)
- ***A****vailability* – The degree to which the product or technology is commercially available
- ***C****apability* – The functional fit (or misfit) between the product or technology and the requirements of the system
- ***T****imeframe* – A measure of how the lifecycle of the product or technology matches the lifecycle for the system. Will it be available when needed? Over the life of the system?

This attribute set appears to address the orthogonality and criticality issues, as well as the issue of "software decay" by providing a measure of not only when the technology or product will become available, but also when it will cease to be viable in a given context. Definitions of each of these attributes are provided in the Appendix. The following sections will discuss an approach to reason about and quantify the time-varying contributions of the individual components of product and technology readiness.

2.3 Acquisition and Development Maturity Growth

To better understand how different types of acquisitions, or different phases in a system development, are affected by the different attributes of technology or product maturity, a picture may be useful. In their paper, Hanakawa, *et al*, model the growth of knowledge during software development, and show how that growth can be represented by the family of sigmoid (s-shaped) curves, as shown in Fig. 1 [9]. Extending this model to a software-intensive system acquisition or development, and equating "knowledge" with some measure of maturity (such as requirements satisfaction or technical performance measure improvement), some generalizations can be made. For example, a typical acquisition or development program will mature slowly during initial concept exploration and technology development, until some critical point is reached (e.g., fundamental science is understood, algorithms validated, etc.) at which time the rate of progress increases. As a program moves towards greater maturity, and most—though probably not all—requirements are satisfied, progress tapers off. In Hanakawa's model, the exact shape of this curve is dependent on the statistical distribution of tasks (e.g., requirements to be satisfied, program milestones, etc.) and their degrees of difficulty, the knowledge/competence of the organization to perform these tasks, and the rate at which knowledge accumulates by performing these tasks. Thus, every acquisition or development will result in a unique "maturity profile."

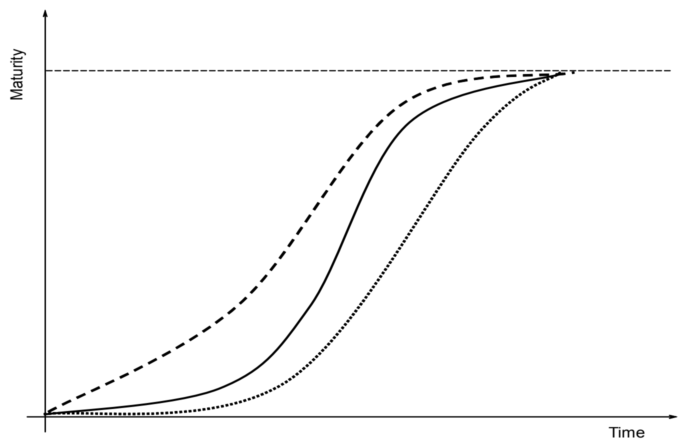

Fig. 1. Characteristic maturity growth curves, or "maturity profiles"

In Fig. 2, typical DoD acquisition phases and milestones are overlaid on a representative maturity profile. Within each of these phases, the slope and curvature (concave or convex) of the curve reflects the changing rate of maturity growth.

This maturity profile provides some insight into how the relative importance of the individual components ImpACT can change during the course of the program. For example, during a technology demonstration (i.e., ATD, ACTD), the fact that a software technology or product is projected to become unsupported sometime during the demonstration is probably less significant than the same situation during the later stages of development, or during post-deployment sustainment. The key to this approach is that, while the absolute values of the individual contributors to product or technology maturity cannot be defined, it is possible to articulate the importance of any one aspect (e.g., "Capability") relative to another, using "fuzzy" definitions like "as important as," or "much less important than" for each phase in a program's life cycle.

2.4 Reasoning Framework

Since ImpACT is defined by multiple criteria, it seems reasonable to use a technique like Saaty's Analytic Hierarchy Process (AHP) to assess the results [10]. AHP defines a process for evaluating multiple criteria, using a hierarchical structure (i.e., goal, attributes and sub-attributes, and alternatives) and pair-wise comparisons to determine the alternative that best satisfies the desired goal. The use of relative rankings, such as "*x* is much more important than *y*" or "*x* has roughly the same importance as *y*," works well in the context of software-intensive system acquisition and development where absolute criteria don't exist, but relative rankings *can* be defined with some de-

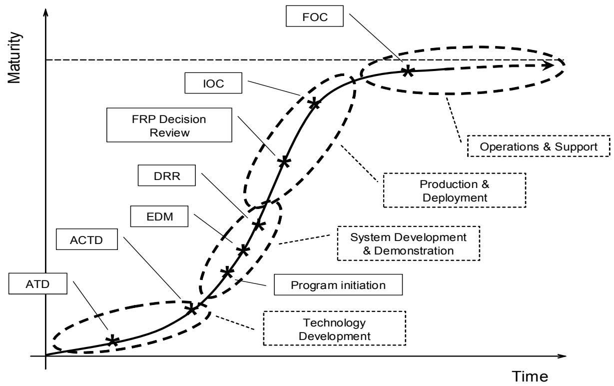

Fig. 2. Representative DoD acquisition/development program phases and milestones overlaid on a typical growth curve

gree of confidence. This approach also lends itself to examining the effects of time-varying affects of the contributions to maturity of each of the ImpACT criteria.

As a simple illustration of using AHP, consider an evaluation of two COTS products ("A" and "B") being considered for incorporation into a system. During this particular phase of the system's development, the criticality of the product to the system (Imp) is considered slightly more important than the ability of the product to satisfy the full range of requirements (C). Both of these attributes are significantly more important than the commercial availability of the product (A), and the temporal availability of the product with respect to the system (T) is of still less concern. Expressed in a pair-wise comparison matrix, this becomes:

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

The two products were evaluated against the ImpACT criteria, with the result:

$$A_{IACT} = [0,2,2,1] \qquad B_{IACT} = [2,0,3,2] \quad (2)$$

Looking at the Importance criterion (I), it was determined that for this specific development, during this life cycle phase, the mapping from ImpACT criteria to AHP relative rankings was as follows:

$$\begin{array}{rcl} \text{ImpACT } 0 & = & \text{AHP } 1 \\ 1 & = & 2 \\ 2 & = & 3 \end{array} \tag{3}$$

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

$$\begin{array}{cc} & \begin{array}{cc} \mathsf{A} & \mathsf{B} \end{array} \\ \begin{array}{c} \mathsf{A} \\ \mathsf{B} \end{array} & \begin{bmatrix} 1 & 3 \\ 1/3 & 1 \end{bmatrix} \end{array} \tag{4}$$

By a similar process, the PCMs for both products with respect to the remaining ImpACT criteria. Applying the AHP method results in weighted scores for the two products as shown:

$$\begin{array}{l} \text{Product A} = 0.68 \\ \text{Product B} = 0.32 \end{array} \tag{5}$$

So, for this example, Product A’s technology readiness appears to be significantly greater than that for Product B, given the relative importance of the readiness criteria (i.e., ImpACT) and the determination of the relative weighting of the values within the criteria.

While AHP provides a method to reason about the contributions of various attributes to satisfying a desired goal, neither AHP nor ImpACT define how the relative rankings of the criteria are derived. Just as the CMM® framework leaves the definition of appropriate processes to the implementing organization, ImpACT leaves the criteria evaluation definitions to the developing or acquiring organization. Within the context of any particular development or acquisition, one approach may be more suitable than another. For instance, there are several possible means of determining the degree of match/mismatch between a candidate product or technology and the desired capabilities; examples include the “Risk-Misfit” approach, described by Wallnau, Hissam, and Seacord in *Building Systems from Commercial Components*, or the “Gap Analysis” methodology described by Ncube and Dean [11, 12].

3 Conclusions

While there is a growing body of evidence that using TRLs as part of an overall risk assessment can lead to an improved understanding of the technological and programmatic risks in a system development or acquisition, there are several difficulties in applying "traditional" TRLs to the evaluation of software technologies. This is especially true for COTS software technologies and products, where TRLs don't provide any meaningful discrimination between mature, commercially-available technologies or products, nor do they take into account the inevitable decay which all software—especially COTS software—experiences. Finally, the existing TRL framework lacks any explicit mechanism to deal with the time-varying effects of the various contributors to technology and product readiness.

The ImpACT methodology provides an alternative to TRLs for COTS software products and technologies which directly addresses each of these shortcomings in a manner which can be tailored to an individual development or acquisition organization, for any software-intensive system.

References

1. Eisman, M., Gonzales, D.: Life Cycle Cost Assessments for Military Transatmospheric Vehicles. <www.rand.org/publications/MR/MR893/> (1997)
2. Mankins, J.: Technology Readiness Levels – A White Paper. <http://advtech.jsc.nasa.gov/downloads/TRLs.pdf> (1995)
3. Department of Defense.: Operation of the Defense Acquisition System). <http://dod5000.dau.mil/DOCS/DoDI%205000.2-signed%20(May%2012,%202003).doc> (2003)
4. General Accounting Office.: Better Management of Technology Development Can Improve Weapon System Outcome. <http://www.gao.gov/archive/1999/ns99162.pdf> (1999)
5. Graettinger, C., Garcia, S., Siviy, J., Schenk, R., Syckle, P.: Using the Technology Readiness Levels Scale to Support Technology Management in the DoD's ATD/STO Environment. <http://www.sei.cmu.edu/publications/documents/02.reports/02sr027.html> (2002)
6. Graettinger, C., Garcia, S., Ferguson, J.: TRL Corollaries for Practice-Based Technologies. <http://www.acq.osd.mil/sis/Conference%20Presentations/TRL%20Corollaries%20for%20Practice%20Based%20Technologies.pdf> (2003)
7. Wong, B.: NASA Cost Symposium – Multivariate Instrument Cost Model-TRL (MICM-TRL). <http://ipao.larc.nasa.gov/symposium/MICM-TRL-Wong.pdf> (2000)
8. Department of Energy.: Modeling and Simulation Technologies Future Combat System Workshop. <http://www.amso.army.mil/topic/fcs/feb-conf/overview.ppt> (2000)
9. Hanakawa, N., Morisaki, S., Matsumoto, K.: A Learning Curve Based Simulation Model for Software Development. <http://ieeexplore.ieee.org/iel4/5475/14745/00671388.pdf?isNumber=14745&prod=CNF&arnumber=671388&arSt=350&ared=359&arAuthor=Hanakawa%2C+N.%3B+Morisaki%2C+S.%3B+Matsumoto%2C+K.%3B> (1998)
10. Saaty, T.: The Analytic Hierarchy Process. McGraw-Hill, New York (1980)
11. Wallnau, K., Hissam, S., Seacord, R.: Building Systems from Commercial Components. Addison-Wesley, Boston (2002)
12. Ncube, C., Dean, J.: The Limitations of Current Decision-Making Techniques. In: Dean, J., Gravel, A. (eds.): COTS-Based Software Systems. Lecture Notes in Computer Science, Vol. 2255. Springer-Verlag, Berlin Heidelberg New York (2002) 176-187

Appendix: ImpACT Attribute Definitions

Table 1. ImpACT Attribute Definitions and Values

Value	***Attribute***			
	Importance	***Availability***	***Capability***	***Timeframe***
0	At least one suitable alternative can be "plugged-in" with minimal tailoring	Widespread commercial use; "shrink-wrap"	"Perfect" fit between component capabilities and system needs	Available over projected life of the system
1	At least one suitable alternative available; reintegration required, with minimal software changes required	Limited commercial use	All necessary functionality present, but some minor "fit" issues	Available when needed, but anticipated that it will be replaced/retired during the system lifetime
2	Moderate reintegration effort required, with pervasive software changes	Public testing (release candidate, beta)	Deficiencies in one or more second- or third-tier functionality; work-arounds available	Available when needed; end-of-life with replacement announced
3	Significant architectural and/or implementation changes required, but extent limited to a single portion/aspect of the system	Private testing (alpha, beta)	Deficiencies in one or more second- or third-tier functionality; no work-arounds available	Available when needed; end-of-life without known replacement announced
4	Significant, pervasive architectural and/or implementation changes required - consider refactoring	"Opportunity-ware"	Significant deficiencies in one or more critical functional areas; work-arounds can provide required functionality with degraded capability	No longer available by "need date," but end-of-life with replacement announced; not available by need date, but alternate product (e.g., prior version) available
5	No workaround available; "back to the drawing board"	"Vapor-ware"	One or more critical functions missing; unsuitable for use	No longer available by "need date," and end-of-life without replacement announced; not available by need date, and no alternate product available