# Technology Readiness Assessments

*featuring Michael Bandor interviewed by Suzanne Miller*

**Suzanne Miller**: Welcome to the SEI Podcast series, a production of the Carnegie Mellon Software Engineering Institute. The SEI is a federally funded research and development center at Carnegie Mellon University in Pittsburgh, Pennsylvania. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

My name is Suzanne Miller, and today I'm very pleased to introduce you to Michael, a senior engineer on the Military Services Team in the SEI's Acquisition Support Program, which we fondly refer to as ASP. Prior to joining the SEI in May of 2005, Mike spent almost 23 years of active duty service in the U.S. Air Force as an enlisted software engineer. In today's podcast, Mike will be discussing technology readiness assessment. His presentation is aimed at engineers who are assigned to an independent review team, an IRT, as well as the program offices that are directed by the DoD, Department of Defense, to have a TRA, a technology readiness assessment, performed on their program. So Mike, what exactly is a technology readiness assessment?

**Michael**: The DoD, Department of Defense, defines a TRA as a formal, systematic, metrics-based process and accompanying report that assesses the maturity of critical hardware and software technologies to be used in systems. It's conducted by an independent review team, an IR team like you stated, which is comprised of a group of subject matter experts. Currently the Department of Defense policy DoDI 5000.02, Operation of the Defense Acquisition System, directs that TRAs are to be performed. This guidance that is provided falls under the assistant secretary of defense for research and engineering. That guidance currently consists of the TRA desk book, dated 2009, plus supplemental TRA guidance that was issued in May of 2011 further refining the process.

**Suzanne**: So in the acquisition system within the DoD, there are lots of different reviews at lots of different acquisition milestones; how is this TRA type of review different from things like a PDR [preliminary design review], a CDR [critical design review], or other kinds of milestone-decision reviews?

**Michael**: Right. This type of review, sometimes people think, "Well, you're just doing a documentation check." A TRA is not a documentation review. There is a lot of planning that goes into it, six months to a year's worth of planning out front. You actually get into design details, engineering studies, test reports, etc. It's really a heavy-duty, engineering level review.

**Suzanne**: So why is it that in addition to all the other kinds of acquisition reviews the DoD feels like it needs this additional review of technology itself?

**Michael**: Basically there are certain sticking points when you are building systems that will trip you up, and the DoD is trying to make sure that these technologies or critical technologies are to the point that they are mature enough. We don't want to make a weapons system that has something with "unobtanium" or something silly in it, that you can't make it, and it makes the cost go out, the schedule go out, the performance is iffy. So they hit on these critical technologies to make sure they are mature to the point when the decision is made to go ahead and pursue the production of the platform or the weapon system.

**Suzanne**: So are there lots of these critical technology elements?

**Michael**: It can vary from program to program. I've seen programs with as few as one or two; I've seen as many as nine. It varies greatly on the domain and the type of system being implemented.

**Suzanne**: That seems to be the sort of nut of the TRA; that you've got to figure what your critical technology elements are and, to paraphrase, I'll say "Are they safe to use in the program?"

**Michael**: Right. You look at things like "Has it been done before?  Has it been used before?"  If you have something, let's say—within enterprise resource planning programs, they usually have what's called an enterprise service bus, some sort of data backbone. That domain, it's pretty common to see. But if you see something inside of a weapons system or a satellite that would be new and novel to that domain, okay, you want to make sure it's going to work for that intended purpose.

**Suzanne**: So it's not just "Is it something brand new?" It's "Is it something new to the environment, new to that domain" as you say.

**Michael**: Yes.

**Suzanne**: That can also make it a critical technology element.

**Michael**: That can also trigger a critical technology element that has to be managed. TRAs are a matter of not only managing the technology, but managing the risk to the program.

**Suzanne:** So the technology risk to the program.

**Michael**: Yes.

**Suzanne**: Okay. Because there are other kinds of reviews that are supposed to manage the programmatic and cost and schedule risk.

**Michael**: Right. Basically the DoD says you have to be a certain maturity for certain milestones, technology readiness level six for milestone B. And then for milestone C, you need to be at a seven. If you're not at that point, they have to go back and ask, "What do need to do to mature those technologies? Can we really meet this deadline? Or maybe we need to go back and look at alternatives."

**Suzanne**: So when you talk about TRL, technology readiness level six and seven, that's part of a nine-point scheme, right?  Nine levels?

**Michael**: That's part of the nine-point scale, that's part of the technology readiness assessment. They go from one to nine, with nine being the most mature, one being the slightly better, than "wouldn't it be cool if" and put some wiring diagrams or something together. The desk book gets into a whole scale and what's expected. Basically as you go up in the numbers from one to nine, the level of integration and maturity and assembly get to be bigger and bigger, both on the hardware side and the software side.

**Suzanne**: So, the way I've heard it expressed is that you move from the research lab environment, to more of an engineering environment, and then into operations, so sort of three big stages.

**Michael**: A lot of the R&D [research and development] usually goes up to about TRL five. When you turn it over to the operations side is when they want to push it to six and operationalize it; seven, eight, nine at that point. Basically at nine, it's been in use for a while. It's well known and mature.

**Suzanne**: So, how would you know if a technology that you're working with is a CTE? Is there kind of a rubric for questions you would want to ask and things like that that would help you understand, "Is this or is this not a CTE?"

**Michael:** The DoD basically provides six questions for evaluation: (1) Does the technology have a significant impact on the operational requirement, cost, or schedule?  That question has to be a "yes."

The remaining five questions: (2) Does the 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) Or, is the technology expected to operate in an environment or achieve a performance beyond its original design, intention, or demonstrated capability? Those last five questions, any one of those has to be answered "yes." And between the first question and any one of questions two through five being "yes" automatically triggers, "This is a strong candidate to be considered a critical technology."

**Suzanne**: So, when we think about software, and how software by its nature evolves and changes—you know, that modified question—it kind of makes me think that almost any significant technology that's software-related should be looked at as a potential CTE, because almost everything gets modified from its intended use.

**Michael**: Right. Software's a little different critter than the hardware side obviously. Hardware, you've got physical things that are tangible. You can do the breadboard, the brassboard, and then make the printed circuit board, etc. and then go to the next higher assembly. Software, you've got multiple pieces that get into things. Not all software technology is necessarily a CTE. You've got the software architecture that can be something new and novel that's never been done before in that applicable environment. The code itself generally is not, unless they've never done that type of code before.

**Suzanne**: You'd be using a new language.

**Michael**: A new language that has never been run on a certain type of processor; with embedded systems, you'll see that. Certain types of software development techniques; if you've got a developer that would really like to try an agile technique but has never done anything for agile other than web development, and you were wanting them to develop a fighter aircraft. That would be a flag. You might want to look at how they're doing the technology development in that area.

**Suzanne**: So even new practices could trigger a technology?

**Michael:** New practices potentially could trigger a technology. If you are reusing code from a prior implementation, the code now provides a functionality that can be a CTE if they've got to go back and modify it "Are you modifying the architecture?" "How much of the code is being modified?" You get into some tricky areas about at what percent modification does it trigger that "modified-from-prior-use"question.

**Suzanne**: So, one of the things that software's really good at and is used for in many DoD applications, is it is this preferred way of implementing optical, or astro-dynamic, or other kinds of algorithms, orbital kinds of stuff, all kind of algorithms, sort of rubrics for doing something. The software is the implementation method for that.

**Michael**: Software's the implementation there, yes.

**Suzanne**: How does that work, because the algorithm kind of has its own novelty in terms of its idea?

**Michael**: Right. For example, if you were using—let's say you had, like in the space domain, an electro-optical, a satellite payload that had some sort of telescope availability. If there was a new data-compression algorithm, or some sort of imaging algorithm, that algorithm, and we are talking the mathematical representation—not the code itself, not the implementation, the code side of it—that would be a possible candidate critical technical element, because it's never been done before. This is the first-time implementation. If it's been done several times in similar domains, then it's more of an engineering-integration issue at that point. The TRAs dance that fine line between "Is it a critical technology?" versus "Is it an engineering or integration?" problem that they have to address.

**Suzanne**: So, there are other risk methods for looking at engineering and integration risks.

**Michael**: Correct.

**Suzanne**: So, you shouldn't have to use a TRA to deal with all those risks. You really want to limit it to just the new technology risks.

**Michael**: Correct. Correct.

**Suzanne**: Okay. So what's your experience? What are some of the sort of interesting software issues that you've run into? You've participated in many TRAs that involve software.

**Michael**: A lot of them in the past several years.

**Suzanne**: What are some of the most interesting things that you've run into?

**Michael**: The software side, like I said, it's a little different critter than dealing with the hardware. Not all software is a technology. We talked about algorithms separate from the design, the architecture, etc. If you're dealing with embedded systems, the firmware, which is another form of software, can actually be at a higher technology readiness level or a lower level possibly than the hardware it's implemented on. They are distinctly different entities. You have to assess them separately.

The relevant environment—which is supposed to stress the most significant parts of the operational aspect, but not the full operational environment—is different for software than it is hardware. On the software side, you're going to higher-end integration with the components. On the hardware side, you're going to the next level of assembly.

**Suzanne**: So, how do you establish sort of the relevant environment for something like software that's going onto a satellite?

**Michael**: It gets back to the system, the system requirements. You have to look at the key performance parameters. Everything relates back to "key this or that." If it affects that operation performance capability, then you start following that thread through that "If I don't implement this correctly, this function fails or it's severely degraded to the point where we can't do the mission with it."

**Suzanne**: So, this is why you said at the beginning that it takes six months to a year to plan…

**Michael**: If done properly, yes.

**Suzanne**: … one of these TRAs because you're looking at all kinds of factors in terms of "Is this a critical technology element?" and then "If it is, is it operating in a relevant environment or just a lab?"  You've got to establish all those parameters that are unique to each program.

**Michael**:  Right. As an IRT, an independent review team, we will establish the criteria for the relative environment if the program office has not already bounded it. In one case we had the program office bound it, and we further refined within those boundaries what we expected to see, the type of tests we expected to be performed and, to the degree of detail, certain environmental factors. Okay, "If the target processor is this processor family, are you at least testing in that same processor class?  Or are you simulating? Is that simulation of a high enough fidelity that it meets the criteria? Or if you can't get the flight article how do you go about testing it on something that you don't have the hardware for?" Things like that.

Some other things that tend to trip people up or programs, we talked about modifying software from prior use a little bit.

With some of these longer-lived programs in the Department of Defense now that go 5, 10, 15 years, if you've got a capability that starts in one increment but completes in a latter increment, where do you assess the technology risk?  Because you can have something that might not meet a threshold in an earlier increment but it's too late by the subsequent increment, so you have to sort of keep tabs on things and address how do you want to take it all or nothing, etc.

One of the more recent ones that we encountered is what I refer to as an "induced CTE," particularly when we go into virtualization, cloud computing, etc. where you've architected a system that's assuming there's going to be physical servers somewhere, and there's certain performance tradeoffs. Now you've got something that's been virtualized, but it wasn't a requirement. Now it changes the architecture. You have now just induced a critical technology element that was outside of your span of control because someone is hosting it on your behalf. And that wasn't part of the design criteria. Things like that you can run into. And that's one of the things we're starting to see.

**Suzanne**: So this is probably not something you want to do without a good bit of guidance. You mentioned at the beginning that there is a desk book that's sort of the authoritative source. Where would people go for that, and are there any other resources you would recommend that they look at if they're involved in either selecting a CTE or performing as part of an IRT?

**Michael**: Basically, like I said, the current desk book is dated as of 2009. And you can find it if you go out the website of the Assistant Secretary of Defense for Research and Engineering. And the website will be available on the website here, the URL.

They published some guidance that came out in 2011 that was part of their efficiencies initiative; they're streamlining the actual process. It doesn't necessarily change the approach to the TRA, but it changes the reporting stream and at what level who requires it. Basically if you're a major defense program, major defense acquisition program designation, you're supposed to have one. If you're an IT-type system, then it's kind of an optional, but it doesn't waive the requirement for the programs to manage that risk and the program manager's responsibilities.

**Suzanne**: Mike, thank you so much for sharing your insights on technology readiness assessments with us today. This podcast is available on the SEI website at sei.cmu.edu/podcasts as I said before. It is also available from CMU's, Carnegie Mellon University's, iTunesU site. As always, if you have any questions please don't hesitate to email us at info@sei.cmu.edu. Thank you very much.

**Michael**: Thank you.