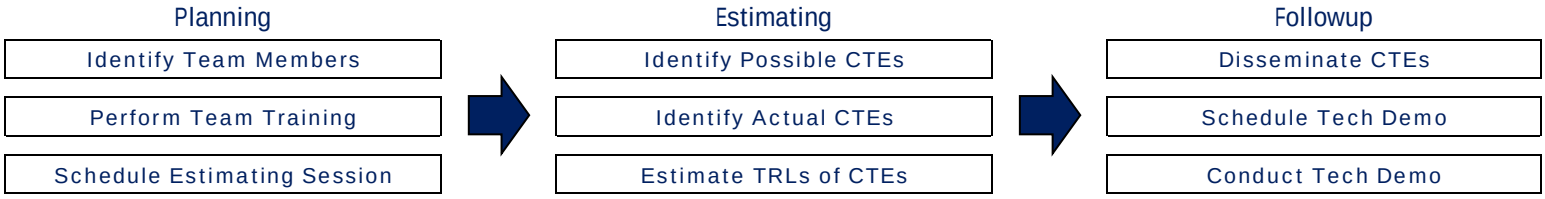
TRA POKER—TECHNOLOGY READINESS ASSESSMENT (TRA) PROCESS FOR TIER 3 ACQUISITIONS (*Using Agile Methods*)

Purpose. The purpose of "*TRA Poker*" is to serve as a lightweight agile Technology Readiness Assessment (TRA) process for Tier 3 acquisitions (i.e., *small projects with total development, test, and evaluation costs under $5 million*). TRA Poker is a variation of "*Planning or Scrum Poker*" used for estimating the size or complexity of user stories (i.e., *system or software requirements for agile projects*). Planning or Scrum Poker is a variation of Delphi (1944), Program Evaluation Review Technique or PERT (1957), Wideband Delphi (1981), and XP Planning Game (1998), which are manual estimating techniques based upon consensus of diverse teams of subject matter experts (SMEs). TRA Poker is designed to rapidly, efficiently, and effectively identify Critical Technology Elements (CTEs) and estimate their Technology Readiness Levels (TRLs) in minutes or hours during the planning stage of Tier 3 programs by Project Management Office (PMO) team members (*versus full-blown multi-million dollar, multi-year independent review teams following DoD 5000-compliant TRA processes*). The value of "TRA Poker" is to improve overall Tier 3 acquisition success by allocating attention and resources to mitigating the sources of technology maturity risk.



- **Identify Team Members**. Select a small three to five-person team consisting of government and contractor technical SMEs. These may include technical directors, lead engineers, architects, developers, computer programmers, and other key technical personnel. (*SMEs with technical hands-on system, hardware, and software development roles are best suited for this task*.)
- **Perform Team Training** (*optional*). If necessary, train SMEs on how to properly conduct TRA Poker. Develop a small set of guidelines, processes, procedures, checklists, self-paced training slides, and/or computer-based training modules on how to conduct TRA Poker. Ask SMEs to review the process and confirm they are comfortable with it. Conduct a mock session in advance to help orient SMEs and calibrate the consistency of their estimations.
- **Schedule Estimating Session**. Identify a meeting location, schedule a mutually-agreeable meeting time, develop an agenda, and hold a highly-structured one or two-hour TRA Poker session. One or more meetings may be necessary depending upon the size, complexity, and cost of the Tier 3 program. One meeting should be sufficient (*and no more than two to three sessions should be required in-total on average*).
- **Hold Estimating Session**. Hold a facilitated TRA Poker session to identify and estimate the TRLs of CTEs. The facilitator should also be trained in TRA Poker and facilitating techniques to ensure the most efficient and effective process. A SAFe Agilist, ACP, CSM, CSSB, PMP, CSEP, or other formally-trained and certified lean or agile facilitator may be best-suited for this role. (*The facilitator is only a process coordinator and does NOT participate in identifying CTEs or estimating TRLs directly*.)
- **Identify Possible CTEs**. Review Product Backlog containing enumerated list of key project epics, features, and user stories (i.e., *system or software requirements*). Identify a short-list of 7 to 15 Technology Elements (TEs) using Brainstorming techniques, which are key drivers, enablers, bottlenecks, and building blocks critical to the success of Tier 3 programs. Vote on the priority, criticality, significance, risk, and impact of TEs and down-select them to a list of 3 to 5 "*Possible CTEs*."
- **Identify CTEs**. Hold a semi-structured round-robin voting session and ask each team member to individually yield a composite answer of "*Yes*" or "*No*" the six (6) CTE questions. "*Yes*" indicates an affirmative response to the first question (i.e., *significant impact on technical, cost, or schedule performance*). "*Yes*" also indicates an affirmative response to one of the remaining five (5) questions (i.e., *risky, new or novel, modified, repackaged, or performance issue*). If the answers are "*Yes*," then the TE is a CTE and if "*No*" then it is not. If the responses are divided ask for an explanation from one of the "*Yes*" and "*No*" responders (*and re-estimate*). Hold two or three estimating sessions until consensus is reached (i.e., *majority decision on "Yes" or "No"*). (*Postpone further discussions if a composite answer cannot be agreed upon and schedule a splinter meeting to resolve*.)
- **Estimate TRLs**. If the team identified any CTEs, hold a round-robin voting session and ask each team member to individually estimate its TRL using the "*Agile TRL*" table (*see next page*). Ask for an explanation of a single "*high*" and "*low*" estimator (*and re-estimate*). Hold two or three estimating sessions until consensus is reached (i.e., *majority value of TRL*). (*Postpone further discussions if a majority consensus on a CTE's TRL cannot be rapidly, easily, and agreeably achieved*.)
- **Followup Action**. If necessary, the facilitator should send a list of the CTEs and their estimated TRLs to the government and contractor PMO (*along with the TRA Poker team*). Such a list can be used by project managers, engineers, and developers for a variety of technology maturation activities. These may include conducting Spikes to reduce the risk and increase the TRL of CTEs (i.e., *rapid sequential or parallel prototyping iterations, Scrum Sprints, or small-scale activities or tasks*). CTEs with sub-TRL 6 or 7 values may simply be tracked using risk management processes, placed in earlier or later iterations or Sprints, or simply removed from the system architecture. In the worst case, the Tier 3 acquisition may be re-planned, delayed, or cancelled if CTEs with low TRLs cannot be matured or mitigated (i.e., *they are so critical and immature that the program cannot proceed successfully or satisfactorily meet its most basic needs, goals, or requirements*).

SIX (6) CTE QUESTIONS ASKED OF SYSTEM TECHNOLOGIES (*from TRA Deskbook*)

1. *Does the technology have a significant impact on an operational requirement, cost, or schedule?*
2. *Does this 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. *Will the technology operate in an environment or performance beyond its original design intention or demonstrated capability?*

"*DRAFT*" AGILE TECHNOLOGY READINESS LEVEL (A-TRL) DEFINITIONS

TRL	Definition	Description	Supporting Information
1	Basic principles questioned, explored, and investigated	Lowest level of technology readiness. A new, advanced, or enhanced system or software technology is investigated by research scientists, engineers, and innovators. Examples include initial enterprise, mission, or customer needs and possible solutions.	• System feasibility plan • Initial product backlog • Initial estimates done
2	Technology concept and/or application examined, studied, and formulated	Once basic technological alternatives and solutions are identified, creation and realization of practical applications can be planned, started, and executed. Examples may include conceptual models, wireframes, or paper prototypes.	• Whiteboard model • Brainstorming results • Conceptual designs
3	Analytical and experimental proof of concept of critical functions and characteristics	Active R&D is initiated. The level at which operational feasibility is demonstrated with physical hardware and software models. Examples include initial hardware and software spikes, iteration-zero activities, and other preplanning activities before actual iterations begin.	• Exploratory code spike • Iteration-zero results • What-if code builds
4	Larger functionality, operational slice, or mission thread validated using laboratory simulations	Basic system hardware and software technology components are integrated to establish that they will work together. Evaluation is limited to essential operational or functional testing, versus the full range of non-functional tests. Examples include an essential system function, slice, or mission thread, feature, or key characteristics of multiple features.	• One or more iterations • Code slice or thread • Up to one code release
5	Function, slice, or thread evaluated using passive or active operational data	Level and fidelity of system hardware and software technology maturity increases significantly. Technologies move well beyond conceptual, feasibility, and operational stages, and are ready for functional and non-functional testing allocated to them. Synthetic and/or other laboratory data may still be in use for test and evaluation purposes.	• One or more releases • Feature slice or thread • Feature tests passed
6	Critical system features developed and evaluated using passive or active data	Representative system-level model or prototype containing most if not all of one or more critical system features satisfying high-priority customer or end-user needs and business or mission value. Features satisfy critical functional and non- functional tests using simulated or passive operational data.	• Main features released • System slice or thread • Multiple code releases
7	Critical system features, functions, and capabilities evaluated using operational data	System-level model or prototype demonstrating most user needs associated with one or more critical system features. Features satisfy most functional and non-functional tests allocated to them, preferably using live operational data.	• Most features released • Release plan on-time • Development tests pass
8	System features, functions, and capabilities completed, validated, and delivered for use	System representation satisfying all functional and non-functional requirements allocated to a backlog and given release plan. Features satisfy all fw1ctional and non-functional, as well as final acceptance tests using live operational data.	• Product backlog done • Release plan complete • Acceptance tests pass
9	[*Legacy*] system undergoing performance tuning, refinement, enhancement, and extension	One or more systems, branches, or variants created to optimize system performance, fix or address critical operational faults and failures, or provide significant enhancements and block upgrades to satisfy emerging market, enterprise, or mission needs.	• Multiple release done • Have operational data • Performance tuning