|  |
| --- |
| **Facilitator Name:** |
| Table Topic: Planning the Lifecycle |
| 1. What does/should "continuous SETR" mean? Possible ideas:    1. Each review is a semi-extended amount of time in which the govt reviews models and digital artifacts with the contractor, culminating in approval of the design to proceed to next phase, or    2. Agile/iterative design where each sprint/epic contains its own "mini" review for the design aspects addressed within that sprint/epic, and eventually all the "mini-reviews" grow together into full design review, or    3. Something else?   *Definition:*  Simultaneously incorporate feedback by reviewing artifacts at defined levels and displayed metrics based on define set criteria on a digital dashboard for decision making  *Important notes:*   * It is very hard to define since it is constantly changing * Breakdown to the smallest possible level and display it on the dashboard that is defined as the program is established * Automation is key to make this happen- it will allow for the simultaneously incorporate feedback, nothing is untouched for more than 48 hours and gives a set of check and balances/ verification and validation * Gives the ability to do reviews as needed- and criteria is established on a case by case basis   *Post it notes:*   * Nothing is untouched > 48 hours * Simultaneous incorporated feedback, deploying feed-forward knowledge, direct application, and knowledge accumulation * Agile = not rigid * Have cadence of event driven reviews but other events can trigger * Set time limit * Ability to conduct review of system functionality at anytime during the system lifecycle * System model are aligned and integrated at multiple abstraction levels * Agile design process that constantly reviews systems * Digital Dashboard display system functionality development with query-ability * Incorporate all relevant stakeholders * Leverages automation for routine, repetitive tasks * Reviewing models; reqs, architecture, simulations against a set of criteria as the system is being developed * Breaking down the system design review to the smallest possible level component * Ability to review/comment on system model designs as the design decisions are being made * Criteria that establishes technical baseline * SETR continuous * Review Format allows for self checking when changes to a system are proposed  1. Right now, SETR is triggered when the contractor delivers specific documents for govt to review, provide feedback on, and then the design is approved via giant meeting when the related documents have been finalized. New ways to trigger SETRs may depend on answer to Q10.1 above. What are pros/cons of these ideas?    1. Similar to draft tech order reviews, do 30/60/90/100% tech design reviews (this is more event-driven when you arrive at that percentage of the design being ready for review)    2. Built-in part of the sprint/epic structure (sort of more schedule-driven, which may be less good than event-driven)    3. Are there other/better ways to define what triggers an event and how it's done continuously?   *New ways to Trigger:*  Triggers begin once there is an “Authority to Proceed”   * Then stakeholders will define criteria and review criteria   + Include the ability to do reviews as needed * Continuous reviews will end when the system retires * There needs to be a review during every criteria defined above. Reviews where we check the artifacts against the criteria   *Pro/Cons*   * Pros:   + Check compatibility (maybe through automation)   + Provides Continuous oversight * Cons:   + Scope Creep   + Uneven time metrics (some thing make take longer than expected)   + Doesn’t account for ‘learning’   *Other thoughts:*   * + Strong feelings that there should be no triggers   + You should be doing a review when a stakeholder requests it, constantly comparing criteria to current, and automation to check against SETR criteria with a minimum of doing a review every 48 hours (either through automation or manually)   *Post it notes:*   * Have a legend status (not started, not satisfied, ready for review, complete) * When I look at the dashboard * As soon as model artifacts are created * Defined metrics of risk trigger a review * Triggers recur every 48 hours comparing SETR criteria to current artifacts * Using continuous SETR requires no trigger * Closure of the previous review (if its continuous the review should start as soon as the last one is complete) * Any change proposed from current iteration * ‘No trigger’ for continuous SETR; it is continuous * Each design stage functional, physical, architecture   + Con: Traditional way similar to CDR, PDR   + Pro: Reduce Risk * Stakeholder decision points trigger capturing metrics of checks against SETR criteria  1. Current Risk Reviews occur during SETRs and PMRs. How should this model change to encourage continuous Risk entries/modifications and reviews along with continuous SETRs    1. What are the risks/ benefits of having a "running" list of risks    2. How/when is a risk added if done continuously    3. What are standard, key metrics that we should track to inform development of new risks?   *Risk Entries should be entered:*   * When a metric Threshold requires a risk input or flagged by automation (or scoring), and/or agreed upon time update for programmatic risks * Every ‘learning’ event should be reviewed as a potential risk * *Each artifact needs a risk and confidence score*   *Other Thoughts:*   * No other thoughts/post it notes due to time constraint |