Open Discussion on Project Planning

# Planning in an Agile Environment

**Key Tenets**

* Planning should focus strongly on the near term
* Eliminate waste caused by planning for eventualities that never come to pass
* Streamline processes to enable rapid and frequent delivery of capabilities

**Planning DOs and DON’Ts**

* DO establish some high-level planning, requirements, processes, and structure; however, activities focus on what teams can develop and field in the next few releases.
* DON’T make detailed plans beyond a program’s ability to control or accurately predict future circumstances
  + Agile methodology does not force programs to establish their full scope, requirements, and design at the start, but assumes that these will change over time. Even so, the program must maintain a big picture, long-term view and focus on the next six-month release.
* DO epics and user stories to concisely define the desired system functions and provide the foundation for Agile estimation and planning.
  + They describe what the users want to accomplish with the resulting system. User stories help ensure that users, acquirers, developers, testers, and other stakeholders have a clear and agreed-upon understanding of the desired functions. They offer a far more dynamic approach to managing requirements than large requirements documents.
* DON’T treat planning as a one-time up front activity
  + In lieu of CDDs and CPDs, programs can develop Requirements Definition Packages (RDPs) to capture a subset of the IS ICD scope and/or Capability Drop (CD) documents for smaller items such as applications
  + Replace comprehensive Preliminary Design Reviews (PDRs) and Critical Design Reviews (CDRs) with more frequent and incremental design reviews during the release planning phases. To demonstrate functionality and provide insight into the program’s progress, these reviews should focus on the relatively small scope of a release and how it aligns to the enterprise architecture. Similar technical reviews can be decomposed to the release level.
* DO continuous planning at every level

# Systems Engineering Role

Program leaders must encourage systems engineers to engage developers, testers, users, and other stakeholders in their disciplined engineering processes. (No ivory towers)

While Agile systems engineering involves frequent informal technical and programmatic reviews, this less formal approach does not equate to less rigor. Instead, greater frequency allows key decision makers and other stakeholders to become more familiar and comfortable with processes in the Agile environment, which enables a more collaborative and productive review process.

* Provide information to all key stakeholders on a consistent, regularly scheduled basis, either through design reviews or program reviews.
* Use the release planning and sprint demonstrations as opportunities to bring users, developers, and stakeholders together in face-to-face sessions to drive collaboration and strengthen teaming arrangements.
* Ensure that once a clear architecture is in place, systems engineers continue to refine it as they learn more from the development sprints and releases.

# Cost Estimation

* Cost estimation in an Agile environment is challenging
* The challenge within DoD is often a resistance to allocate budget for a program until all the requirements are fully defined and approved. There needs to be a clear understanding of the level of requirements maturity required for budget authorizations.
* One benefit of Agile is that once a budget has been established the program can be structured to “build to budget.” The funding that the program receives then drives the number of releases it can manage in a given year and the totality of delivered requirements within the entire development period of performance.
* During the program execution phase, a high-level program estimate undergoes refinement to create detailed release and sprint-level estimates as requirements become better defined. The fidelity of the cost estimate increases once a development team is established to help estimate the level of work for each requirement (i.e., as translated into user stories)

