Industry Panel

How Can Systems Engineering Help Improve Program Development and Execution in Times of Tight Budgets?

Paul Bailey
Chief Systems Engineer
Raytheon Company

NDIA 14th Annual Systems Engineering Conference
29 Oct 2013

Perspective

- Looking through the lens of a Chief Engineer
  - Program and Business
- A lot of time spent fighting fires
  - Program schedule and technical difficulties
  - Test Failures
- In my experience, the vast majority of fires are started by inadequate front-end systems engineering

Just Two Examples…

- Integration of a Radar into a System-of-Systems
  - Problem: Attempt to develop Interface Specifications without having created a CONOPS and true functional allocation – *"Just modify the existing ICD"*
  - Outcome:
    - ICD and IRS took an interminable series of meetings to develop, delaying software development and testing – *"Hmm, what might we use this field for?"*
    - Desired functionality lacking – *"What do you mean I can't drop a track?"*
- Integration of Existing Components into a Weapon System
  - Problem: Integrate the components without end-to-end analysis capability to support architectural trades and functional allocations – *"Already been done for all the components"*
  - Outcome:
    - System performance lacking – *"Less than the sum of the parts"*
    - Test preparation capability inadequate – *"You assumed what?"*

Now More Than Ever

- Why this keeps happening
  - Hard to quantify the benefit of avoiding future problems, rework, etc.
  - Leveraging existing systems makes it easy to assume away the need
  - In a cost constrained environment, SE can be a tempting target for perceived cost reduction
- Why it *can't* happen any more
  - Programs can't afford delays, rework and overruns due to inadequate or incomplete systems engineering
  - The DoD can't afford to buy systems that are not all they can be
  - The Warfighter can't afford the risk of system failures