New Roles for Architecture

Christopher Beres, Maj/USAF
Christopher.Beres@losangeles.af.mil
Agenda

- An architectural vision
- What we are doing
- What we are planning
- Current observations
- Guidance recommendations
Caveat

- Talk presents a vision and approaches that suggest new roles for architecture
- Any insights and recommendations are lessons learned
Definitions

_Architecture_ is the organizational structure of a system or component \(\text{(IEEE Std 610.12-1990)}\)

_Architecture_ is the underlying abstraction that encompasses all the requirements of the system \(\text{(Dr. Jaime Milstein)}\)

_Architecture_ is the means of representing a communication relationship between those who want a system and those who build it \(\text{(Maj Christopher Beres)}\)
An Architectural Vision – a major paradigm shift

- View architecture representation as a means to
  - Understand requirements as part of the architecture
  - Discover and manage software risks due to complexity
  - Reduce program risk and ensure mission success prior to design and code implementation
  - Plan the evolution of our assets

- 1. Architecture as blueprint:
  - Architectural analysis needs to be a precursor to the design and code implementation
  - Think less as an end-product and
  - more as a means for evolving insight throughout the design and code implementation
  - Better insight means earlier resolution, lower cost/schedule, lower risk
  - Coherent architectural commitment is essential, expected, and must be planned at the early stages of the program
What we are doing

- **4. Architecture representation**
  - Developing automated analysis tools, e.g. Real-time Embedded Architecture-Centric Testbed (REACT) to achieve early insight into architecture problems
  - Analyzing contractor-provided architecture artifacts
  - Improving our representations to support evolution

- **6: Architecture as basis for requirement verification**
  - Performing static and dynamic Unified Modeling Language (UML) analysis
  - Verifying requirement allocations and mapping to use cases

- **9: Architecture as a tool to manage change**
  - Representing architectural details is important for evaluating unforeseen lifecycle architectural concerns over its lifecycle.
  - Building tools such as REACT to enable such analysis.
What we are planning

- **7: Architecture as basis for System Testing**
  - Looking at use cases to see how requirement dependencies can improve test case construction.
  - Preparing use case logical flow analysis (e.g. pre/post conditions)

- **8: Architecture as basis for System Implementation**
  - As-built to as-designed architectural differences
  - Use as-built information to refine earlier models and analysis.
Current observations

4. Architecture representation
   - Representation takes many forms: UML models, word docs, spreadsheets, ICDs, etc.
   - Analyzable electronic representation is essential for lower risk

2. Multi-views:
   - Consistent multiple views are hard to achieve in current large programs.
   - Often problems due to mismatch in granularity provided, expected, needed
   - Starting with a bad process leads to a bad architecture (cascades to bad design and bad implementation).

3: Architecture as Decision Making Tool:
   - Early insight requires early response by government and contractors
   - REACT promotes early decision making
   - Need new ways to improve the reporting and handling of early discovery shortfalls
Guidance recommendations

- Mission success requires cooperative collaboration between government and contractors

- Architectures aren’t just delivered—they evolve

- Architectural representation is a means to support this collaboration and evolution

- Architectural granularity should be driven by various goals:
  - Understandable conceptual model of our reusable assets (e.g. legacy compatibilities)
  - Risk reduction studies throughout entire lifecycle
  - Need to capture design flexibility for systems likely to change over time