

**Introduction to
Architecture-Centric Evolution and Evaluation of Software-Intensive Systems
Breakout Group**

**Lt. Col. Laura Pope
Air Force Space and Missile Systems Center**

You've all heard it said, "If you don't know where you're going, any road will get you there." However, I believe that it absolutely does NOT apply to software development. In software, if you don't know where you're going, then NO road will get you there, at least not at any reasonable cost or schedule!

I have over 11 years of experience in managing software-intensive systems – systems with over 1 million lines of code.

An ironic point here is that when I first came into the Air Force in the late 1980's, the big development was the Strategic Defense Initiative (SDI, also commonly called "Star Wars") program. At that time, I remember the largest criticism of the feasibility of SDI was that it was estimated to take 1 million lines of software code, and nobody thought we would EVER be able to build and field such a large software-intensive system....but I digress!

Some of the software-intensive systems I managed were already in operations, being maintained and upgraded, and some were in the development and fielding stages. And I have found that the quality of the acquisition and fielded system directly corresponds to the maturity of the Concept of Operations (CONOP) and the quality of the upfront architecture concept.

In every case, the more clearly my contractors and I understood the different architecture views of the system, the more likely the Govt was to get the product the Government wanted at a cost and schedule we could all live with.

Bottom Line: Why does the Government care about migrating to an architecture-centric evolution and evaluation of software-intensive systems? Because of money, of course. Historically, software-intensive systems have been notoriously hard to predict and budget for final costs. I believe this is due to three reasons:

- 1) The concept of who and how to operate the software-intensive system is not fully mature at the start of software design; therefore, there are understandability issues with how to implement (code) the requirements to suit the actual operations.
- 2) The Government changing its mind about interfaces, adding new requirements and interfaces, changing the CONOP, and general programmatic instability; therefore, there are issues with the flexibility, extensibility and executability of the software baseline.

- 3) The architecture design does not adequately consider operations and maintenance (O&M) costs of software-intensive systems, which leads to issues of Maintainability.

I believe each of these three driving issues, and their associated costs, can be mitigated by creating and documenting architecture-centric views of the software-intensive system upfront, fully coordinating them with all system stakeholders, and keeping them current.

During this workshop, we are going to hear different opinions and recommendations for developing the architecture views I mention above. I'm looking forward to it, and am sure you are too, so let's get started....