Integrating Software and Systems Engineering to Promote Interoperability

As the Acting Director of Software and Systems Engineering in the Office of the Deputy Under Secretary of Defense (Acquisition and Technology), I occupy a uniquely advantageous position to witness the challenges that interoperability imposes on the engineering community. In particular, I have observed that very distinct cultures distinguish the software engineering and systems engineering communities and the subtle differences in their perspectives, and how these have undermined the DoD's ability to develop solutions to warfighter needs. Although systems engineering, as a discipline, nominally encompasses software, its heritage is hardware-oriented and favors a product-oriented perspective and functional decomposition. By comparison, the software community—with roots in embedded systems, information technology, and command and control domains—embraces layered architectures and process-focused development perspectives.

Perpetuation of functionally stove-piped policies and organizational structures has permitted each community's worldview to somewhat peacefully co-exist by limiting the amount of interaction. Today's systems cannot ignore the need to interoperate. Information technology has allowed us to shift the balance of control from hardware to software and enabled an ad-hoc composition of systems which are not specifically designed to interoperate. We face warfighter demand for capabilities that are joint and adaptable. This requires acknowledgement of an enhanced systems engineering imperative to seamlessly integrate hardware, software, and human factors and enable system of system solutions. We must synthesize the most thoughtful perspectives of software and systems engineering to capitalize on technology and deliver integrated capabilities to our customers.

This issue of CrossTalk addresses some of these compelling challenges as we strive to improve integration/interoperability. In *Systems Engineering for Capabilities*, Dr. Judith S. Dahmann, George Rebovich Jr., and Jo Ann Lane adapt systems engineering concepts for the development of capabilities. Dr. John Colombi, Maj. Brannen C. Cohee, and Maj. Chuck W. Turner discuss—in *Interoperability Test and Evaluation: A System of Systems Field Study*—the policy and practice of testing for interoperability in a system of systems context. Shamlan Siddiqi's *Key Transformational Techniques to Achieve Enterprise-Scale Interoperability* details the use of service-oriented architecture design principles and an agile methodology. William B. Anderson and Philip Boxer's *Modeling and Analysis of Interoperability Risks in Systems of Systems Environments* describes how their techniques in an interoperability risk probe found gaps in the ability of a modernization program to react to changing demands. In *Quality and Cost – It's Not Either/Or: Making the Case With Cost of Quality*, George Webb and LTC Nanette Patton describe the application of "cost of quality" principles to permit the balancing of these attributes.

I cannot overemphasize the importance and criticality of the need for the software and systems engineering communities to jointly confront the challenges of fielding systems that are affordable, sustainable, and interoperable.

Kristin Baldwin

Acting Director, Software and Systems Engineering
Office of the Deputy Under Secretary of Defense (Acquisition and Technology)