Best Training Includes Going Back to Basics

I recently had the opportunity to attend a meeting conducted by a major program executive officer (PEO). What was the topic? Training. Here were lead engineers and program managers talking about how to train their people in the basics of project management, configuration management, and measurement. It was refreshing! How did they get there? Upon assuming his new duties, the PEO used several definitive references on software development practices: a text, the Software Engineering Institute's "Guide to the Capability Maturity Model® for Software (SW-CMM)," and several training manuals. After some study, he began an initial review of the programs in his portfolio. During this process, he asked questions on key process areas to those managing each program. The message he received? A lack of attention to the basics.

This PEO is not alone. From Ford Motor Co. and the Air Force's Air Logistics Centers to Marvel Comics, all have formal "Back to Basics" programs aimed at reinforcing the organizations' foundation: properly trained people using correct tools, processes, and data. Total Quality Management, Lean Six-Sigma, Benchmarking, ISO, SW-CMM, Capability Maturity Model® Integration℠ – all these depend on having the basics in place. This month's CrossTalk is devoted to some of the fundamental areas necessary to successfully complete software intensive system development.

In *Overview of Project Management*, Tim Perkins, Roald E. Peterson, and Larry Smith of the Software Technology Support Center (STSC) outline the definitions, skills, and processes required for effective project management. Their work is derived from experience in the field and instructing the STSC's Project Management Workshop. Next, in *Delivering Quality Products That Meet Customer Expectations*, Louis S. Wheatcraft emphasizes the importance of establishing a shared vision of the product at the beginning of the project. He then shares best practices and lessons learned in defining product scope such as stating an implementation as a need vs. understanding the customer's need.

Making Measurement Work by Cheryl Jones provides lessons from successful measurement programs and outlines the Practical Software Measurement framework. The current Department of Defense acquisition environment fosters using performance-based contracts with prime contractors for major systems serving as integrators. This makes it crucial to understand critical software architecture, risks, and team capabilities to know what metrics to build into those contracts. A successful measurement process must be a way of doing business and the basis for making fact-based decisions.

Also from the STSC is *But I Only Changed One Line of Code!* by Theron R. Leishman and Dr. David A. Cook. This one-act play provides motivation for the application of one of the most widely accepted software best practices: configuration management. Although generally accepted, basic configuration management activities are often ignored, resulting in serious negative impact on software development and acquisition projects. This article introduces basic software configuration management concepts and the rationale for its implementation.

In *Risk Management Applied to the Reengineering of a Weapon System*, Claude Y. Laporte and Guy Boucher briefly describe a systems engineering process and discuss the application of risk management practices to the reengineering of operator console stations of a missile weapon system, including 12 lessons learned.

Louis A. Poulin's article, *High Quality, Low Cost Software Inspections*, cites work by Ronald A. Radice in his book, "High Quality, Low Cost Software Inspections," and defines inspections, peer reviews, walk-throughs, and structured reviews. He explains that while these are all terms that are used interchangeably in software engineering, the activities are rarely carried out consistently in the course of developing an application.

Are formal methods the answer to providing high-quality software through mathematical rigor? In *Application of Lightweight Formal Methods in Requirements Engineering*, Vinu George and Dr. Rayford Vaughn explain how using formal methods is important to achieving correctness, consistency, and developmental understanding. However, the degree of formalization must be carefully planned. An evolving approach – lightweight formalism – which is less rigorous than normal, can be advantageous.

A wealth of information exists on each of these fundamental topics. The STSC's Web site, <www.stsc.hill.af.mil>, contains additional references. We hope this collection of articles provides you a quick review that you can use when training your team.

Glenn A. Palmer

Lt. Col. Glenn A. Palmer
Director, Computer Resources Support Improvement Program