Successful Software Is an Epic Production

From conception to completion, good software requires a more-than-just-code mindset. Good software is the product of the successful execution of hundreds of sound project and process practices. The list, which requires strict adherence to, might seem staggering to the casual observer: documented policies and procedures; requirements management; detailed project planning; system design; configuration management; peer reviews; tracking and oversight; collecting and managing with metrics; accurate useable documentation; and thorough testing against requirements at the code, subsystem, and system levels, just to name a few.

At the next level, good software becomes excellent software through institutionalized quantitative processes that are measured, analyzed, and optimized over time. At Hill Air Force Base, Utah, the 309th Software Maintenance Group (309 SMXG) has achieved this level by embracing continuous process improvement. The 309 SMXG has an established Capability Maturity Model® Integration process capability and is adding new capabilities as it strives to implement AS9100 quality management standard requirements.

Yet it doesn't stop there. As important as process is, you can't produce software without people. Successful execution occurs at the hands of trained and cross-trained, experienced, mentored, and motivated people. As systems become larger and more complicated, employees with greater years of experience become even more important. Like Jim Collins said in his book "Good to Great," "...people aren't your most important asset; *the right people* are your most important asset."

Randy B. Hill

Ogden Air Logistics Center, Co-Sponsor

Software Development Is More Than Coding

This month we announce that nominations begin for the 2005 Top 5 U.S. Department of Defense Programs Awards. The Office of the Undersecretary of Defense is once again recognizing those programs most successful at applying systems engineering best practices in the management, development, and integration of hardware and software. This award is in line with CROSSTALK'S theme this month: total creation of a software project. Leaders in the software community are constantly reminding us that good software development involves so much more than developing code. All the players, including customers, must employ sound practices such as effective requirements definition, removing defects at their origin, measurement throughout, effective training, and communication.

We begin our theme section with *Correctness by Construction: A Manifesto for High Integrity Systems* by Martin Croxford and Dr. Roderick Chapman, who discuss how these practices apply to the entire development life cycle. Next, Granville Miller discusses a Microsoft approach to agile software development in *Agile Software Development for the Entire Project*. In *Eliminating Embedded Software Defects Prior to Integration Test*, Ted L. Bennett and Paul W. Wennberg discuss how testing software can be performed during the entire development cycle, even design.

In our supporting articles, Watts S. Humphrey's *Acquiring Quality Software* discusses using measurements to ensure quality, and in *Role of Human Emotions in Requirements Management*, Sreevalli Radhika. T. discusses how a requirements document might affect the entire development process. We wrap up our 18th volume with the Article Index on page 28, highlighting articles published in 2005.

Elizabeth Starrett

Associate Publisher