# Reinforcing Good Practices

CrossTalk would like to thank the Department of Homeland Security for sponsoring this issue.

We all have been taught sound practices since childhood. Remember the ol' dental mantra of "don't forget to brush your teeth after each meal"? Those instructions were soon augmented with a warning that brushing alone was not enough, and that flossing and regular check-ups were needed to reinforce brushing and prevent the development of dental maladies. Our experience with these routines over our lifetime confirms the worth of reinforcing good, basic practices. Even armed with this knowledge, many of us at some point—then, now, or along the way—ignored the act of reinforcement and suffered the occasional, painful cavity.

Software practices are similar because we understand the value of implementing well-defined best practices, code reviews, and well-structured architectural design in combination with the basics. Even with that understanding, the drive for on-time delivery or budget and time constraints hampers the opportunity to perform the reinforcing actions that prevent future problems—and we suffer the software equivalent consequence, otherwise known as software defects.

The March/April issue of CrossTalk provides five well-crafted articles intended to assist developers in avoiding the pain of "software cavities" by bolstering their current processes through implementation of sound reinforcement practices. Dr. Nancy R. Mead, Dr. Dan Shoemaker, and Jeffrey A. Ingalsbe share best software assurance practices as developed by the Ford Motor Company in *Software Assurance Practice at Ford: A Case Study*. D.T.V. Ramakrishna Rao increases developer's defect awareness through a reinforcing practice known as "active reading" in *Defect Detection by Developers*. SEI authors Michael Gagliardi, William G. Wood, John Klein, and John Morley offer a consistent approach for evaluating and mitigating risk and challenges to large-scale systems in *A Uniform Approach for System of Systems Architecture Evaluation*. In *Static Analyzers in Software Engineering*, Dr. Paul E. Black contrasts the strengths of static analyzers with testing as a method for detecting possible code problems. Roger Stewart and Lew Priven explore ways for leaders to make software inspections unassailable in *Management's Inspection Responsibilities and Tools for Success*.

CrossTalk offers two additional articles this month that complement our theme quite well. In *The Evolution of Software Size: A Search for Value*, Arlene F. Minkiewicz, through her own industry experiences, analyses 25 years of efforts to solve the problems of software size. As well, Katherine Baxter offers up her article, *Understanding Software Project Estimates*, in an effort to remind us of the virtues of software cost estimating on each and every project undertaken.

In closing, remember the words of English philosopher and scientist Francis Bacon who once said that it is "... not what we profess but what we practice that gives us integrity." While Sir Francis wasn't referring to software when he made this statement, software integrity and quality can only increase when we follow his advice and reinforce our already good practices.

Kasey Thompson

Kasey Thompson
*Publisher*

## Notice a Few Changes?

Beginning this month and for the remainder of 2009, CrossTalk will be published every other month. CrossTalk issues in May/June, July/August, September/October, and November/December will be larger in volume with additional articles. Our hope is that this format will allow for more detailed articles in a format more conducive for defense software engineering. Also, we will be adding some new elements, beginning with the sponsor logo box at the top of this page. Look for more changes as we begin to incorporate them throughout the next few issues.