



## What's in Your Toolbox?



If you were to start a discussion on software tools, most people's initial frame of reference would probably be tools such as modeling languages, compilers, word processors, project management tools, etc. While these are all important tools, I hope that this issue of CROSSTALK will expand the frame of reference for our readers. When considering software tools that may help with software development and acquisition, projects will realize more benefit if the project team expands its consideration of tools to include helpful processes and techniques in addition to software products such as those

listed above.

Over the years, CROSSTALK has shared many tools that apply to most aspects of software development and acquisition; some examples that readily come to mind include updating legacy code, working with people, information security, architectures, and processes such as those promoted in the Capability Maturity Model® (CMM®) Integration and ISO 9000. These are all great tools for improving the quality of software projects, the efficiency in developing software, and the ability to accurately predict the cost and schedule of delivery. The CROSSTALK staff developed this special issue to highlight the idea that tools for developing software are more than just software products. Some of the most useful software tools are the ones most often neglected by software developers, yet much has been expended over the past several years to educate developers (and now acquirers) about these tools and their benefits.

We begin this issue of CROSSTALK with an article from Dr. Alistair Cockburn that truly stresses this point. In *What the Agile Toolbox Contains*, Cockburn discusses numerous tools from all angles of this discussion. If you're not involved with agile software development, you'll see many of these tools apply to other development methods as well. I recommend this article for everyone.

In *A Revolutionary Use of COTS in a Submarine Sonar System*, Capt. Gib Kerr and Robert W. Miller share the success they have achieved thanks to the use of commercial off-the-shelf (COTS) software. As discussed, this effort was not without its drawbacks, but the benefits outweighed the problems.

Next, Dr. Mikhail J. Atallah, Eric D. Bryant, and Dr. Martin R. Stytz discuss various approaches to anti-tamper technologies in *A Survey of Anti-Tamper Technologies*. This discussion encompasses introductory descriptions of recommended technologies, including their benefits and their drawbacks.

Safety critical software presents additional challenges for the developers. In *Safety Analysis as a Software Tool*, Blair T. Whatcott discusses the basic steps for safety analysis and reminds the readers that safety analysis must be performed at the system level, since many hazards exist at interfaces between system components.

In *Three Essential Tools for Stable Development*, Andy Hunt and Dave Thomas share their experience that configuration management, unit testing, and automation are key to mitigating a majority of the common problems experienced by software developers.

We conclude this issue with a high-level discussion on making measures more useful with David B. Putman's *Your Quality Data Is Talking – Are You Listening?* Putman was one of the key people who helped Hill Air Force Base's Software Engineering Division receive a Level 5 rating on the CMM. In this article, he discusses some of the thought processes that helped so much with the measurement efforts.

You might notice that there are no supporting sections this month. The reason for this is that the CROSSTALK staff believes all of the ideas discussed in these articles should be considered useful tools that support software development. We hope CROSSTALK is also in your software toolbox.

Elizabeth Starrett  
Associate Publisher