CrossTalk would like to thank NAVAIR for sponsoring this issue.

Tools and processes can be very powerful enablers of Software Engineering. The right tools in the hands of capable people who make effective use of those tools can make the difference between project success and failure. Well thought-out, effective processes can help avoid mistakes and costly rework. A good fiddle is important, but it takes a good fiddler to make the best music.

Some of the challenges we face when selecting and using tools and processes are:

- Whether to centralize or distribute their use. For example, code analyzers can be expensive to procure, require time and training to use effectively, and produce a lot of false positives. Having a small, highly-skilled team to perform such code analysis could actually be advantageous to each program office and software support activity doing this on their own.
- Balancing strict adherence to process with fostering innovation and creativity and doing things expeditiously. While we certainly want to do things right the first time, we also want to enable such things as Rapid Deployment Capability (RDC) to get the product our Warfighters need as quickly as possible.
- Keeping our tools and processes up to date and relevant. We work in a very dynamic environment, so should always be looking critically at our tools and processes to ensure they appropriately complement our programs.

As with any resource, tools and processes can be used effectively, used ineffectively or even misused. A sharp saw, as Dr. Stephen Covey pointed out, is easier to use and cuts better than a dull saw. A chain saw is more effective than a hand saw, but also has the potential to do more damage and cause injury. Adherence to process can assure the job is done right but also cause a bottleneck. It is up to us as software practitioners to select the appropriate tools and processes, and use them effectively to get the most benefit.

I hope as you read the articles about tools and processes in this issue, you think about how best to adapt them to your programs.

Al Kaniss
Software Engineering Branch Head
NAVAIR