



## Gold Medal Requirements



At the time we prepared this month's issue, the world was coming together for the 2002 Winter Olympics in Salt Lake City. In case you did not know, the CrossTalk staff is located just 30 miles north of Salt Lake City at Hill Air Force Base. For me, it has been very exciting to be in the midst of all of the Olympic activity. Most of all, I'm happy to see the world come together in peace for the games.

During the past few years, Salt Lake City and its surrounding communities have been transformed with improved highways, sports venues, and new businesses in an effort to support the games. As this month's issue highlights requirements and how to avoid the risks associated with specifying poor or invalid requirements, I could not help but think of all of the software requirements associated with putting on the 2002 Winter Olympics.

I recently attended a conference in which an employee of a local telecommunications firm spoke. He identified a myriad of information technologies (IT) employed around each of the Olympic Venues. From computer terminal rooms and networks for media support, to security support, to athlete housing support, to event-ticketing support, and to the events themselves, the software needs associated with hosting the games is absolutely phenomenal. Specifying good and valid requirements to meet the Olympic IT needs could not have been a more important step in making this world event a possibility as well as a huge success.

The defense software world has its own set of challenges when it comes to gathering, specifying, and validating weapon system requirements. Two of the Air Force's Software Technology Support Center's consultants, Theron Leishman and Dr. David Cook, lead off this month's issue with sound advice in *Requirements Risks Can Drown Software Projects*. In this article, you will find helpful tips on how to recognize requirement risks and how to mitigate these risks to keep software projects from going "overboard and drowning" throughout their life cycle.

Next, Dr. Ralph Young discusses *Recommended Requirements Gathering Practices* and explains how requirements elicitation techniques such as interviews, workshops, and storyboards can ensure effective requirements definition and communication among all stakeholders. And in *Reducing Risks Through Proper Specification of Software Requirements*, Al Florence critiques eleven examples of requirement specifications and suggests an improved and less risky way to state these example requirements.

Lastly, be sure to check out our Software Engineering Technology and Open Forum sections this month for more insight into the many challenges and potential risks that defense software projects face today.

I hope that you find this month's issue helpful with techniques and reminders on how to keep from specifying risky requirements. Take the time to make your software project requirements as good as you can, just as if you were training and competing for a gold medal at the Olympics. May we all be winners in the defense software community.

Tracy L. Stauder  
Publisher