Times, They Are A Changin' for Software Development

It is said that people fear change. However, we embrace change all the time when the results are known. We change what we eat (three meals a day of Wheaties doesn't sound very good). We change what we wear (the same pair of socks all month isn't a pleasant image). It isn't change that people fear as much as the unknown results of change. Have you ever watched a tape-delayed game on TV when your team won? I have. At halftime, when my team was down by 25, I wasn't nervous or afraid at all. Why? I knew the outcome.

This issue of CrossTalk focuses on the changing landscape of software development and some of the new methodologies being pioneered in the software industry. Having been successfully implemented in the commercial software industry, Object Orientation and the Universal Markup Language (UML) present some potential benefits to the real-time and embedded systems industry. From ARTiSAN Software Tools, Alan Moore suggests that some extensions to the UML are needed to address the lack of modeling techniques for real-time and embedded systems.

LTC Michael Bowman, Dr. Antonio Lopez Jr., and MAJ James Donlon from the U.S. Army War College and Dr. Gheorghe Tecuci from George Mason University present a new paradigm for software development. In their article you will discover Disciple, an artificial intelligence approach to the development of knowledge-based systems. Disciple is intended to replace the indirect transfer of knowledge and expertise from the end-user to the development team, and then into the software product with direct user development of the system with the assistance of an intelligent software agent. This is quite a revolutionary approach that may cause you to rethink traditional defense software development methodologies.

In a world where acquisition wants things better, faster, and cheaper, and software engineering requires process, discipline, and rigor, extreme methods may be the solution. From the Software Technology Support Center, Theron Leishman discusses Extreme Programming (XP) and suggests some caveats when implementing this innovative development methodology.

At a time when software process is being touted from every rooftop, Richard Duncan, a master's student at Mississippi State University further suggests that XP be designed for medium- to small-sized organizations. XP, designed with requirements drift as a fundamental occurrence, nominates coding as the key activity throughout the development process. While sounding a bit extreme, Duncan's article provides some thought provoking ideas.

From the Software Engineering Institute, Drs. Stephen Cross and Caroline Graettinger suggest that, given the rapid pace of change in software engineering, software-intensive organizations must develop a core competency for proactive change management. They also suggest that a career path in change management will be a critical need and opportunity for the 21st century software engineer.

Six valuable lessons learned about the acquisition of commercial, off-the-shelf-based systems (CBS) are at the heart of an article from Richard Adams and Suellen Eslinger of The Aerospace Corporation. Adams and Eslinger, having performed an in-depth study of actual CBS, identify some shortcomings of, and potential changes to, the acquisition and development of COTS-based systems.

Change really is a part of the software development landscape. We hope that these articles will help you gain a greater understanding of this changing landscape so that when someone suggests a *new approach*, your first reaction is not one of fear. As always, your comments and questions are welcome.

Kevin Richins
Software Technology Support Center