



Making an Educated Guess



What does estimation mean? Having spent 30 years as an engineer and manager in the electronics and software marketplace, I thought, "Oh, that's easy! You simply *guess* how long it is going to take you to do this job, based on your last experience with the same or a similar job, and then *double* that guess."

Two things are clear with this oft-used algorithm. The first is that to have any kind of chance to be in the ballpark, you need some knowledge or historical precedence for having done something similar in the not-too-distant past. The second is a realization that the accuracy of any such estimate is not very good.

If your business success does not rely on the accuracy of this type of estimate, this algorithm will probably continue to be used. Only when a business' existence and success rely on something a little more accurate is it likely a little more effort will be applied to make cost estimates more realistic and reliable.

As readers will find in this month's *CROSS TALK*, cost estimation has become an essential feature of any successful software business development or sustainment venture. However, as quoted in *Reducing Bias in Software Project Estimates* by David Peeters and George Dewey on page 20, some still believe that "software estimating is definitely a black art." The authors note that a large percentage of software projects continue to finish behind schedule and over budget. The article discusses some ways to identify and reduce biases in software cost estimation to help make estimates more accurate.

The dramatic increase in the quantity and complexity of software that drives so many of today's defense and consumer products is also a major driver in the need to develop more robust estimation methodologies and tools. Barry Boehm and his co-authors write in *Future Trends, Implications in Software Cost Estimation Models* on page 4 that software development trends have turned from the Waterfall process model toward evolutionary, incremental, and spiral models. It was found that the very useful Constructive Cost Model (COCOMO) estimation techniques required some significant changes to keep pace with the changing nature of this software development evolution. Product line management approaches to software reuse, and graphic user interface builder tools that made traditional-size metrics such as source lines of code inappropriate are examples of new software design methods that required new techniques. This article also discusses how continuing development extensions are being made to COCOMO II to address emerging trends such as Rapid Application Development and commercial-off-the-shelf integration.

We hope that readers will find these and the other articles in this month's *CROSS TALK* to be valuable resources as they continually improve their organization's ability to accurately predict project schedule and cost.

H. Bruce Allgood

Deputy Computer Resources Support Improvement Program Director

CROSS TALK welcomes Bruce Allgood, Deputy of the Computer Resources Support Improvement Program (CRSIP) at Hill Air Force Base, Utah. Allgood replaces Lt. Col. Joseph Jarzombek (retired) as the CRSIP sponsor of our journal. As a member of the Air Force Software Technology Support Center, Allgood has supported software process improvement efforts throughout the Air Force and the Department of Defense. He also represents the Air Force Materiel Command on the Practical Software Measurement Technical Steering Group (PSM), the Office of the Secretary of Defense's Software Collaboration Team, and is a certified PSM trainer. He spent 20 years in various management and development roles at leading commercial electronics and software corporations, including 11 years at IBM, two years at Hughes Aircraft, and five years at Hewlett Packard. Allgood received his bachelor's degree in electrical engineering from the University of Utah and a master's degree in electrical engineering from Colorado State University.