



Good Software Estimation Requires Historical Data



This month's CROSSTALK focuses on software estimation techniques used within the industry. These articles all agree that good software estimation requires an understanding of the capacity of any particular organization to deliver a software solution within their specific environment. Such capacity evaluation must be based upon historical contexts and good risk management techniques. These articles also provide a good perspective on the state-of-the-art in software estimation and emphasize the importance of measuring how well estimation is being done through each phase of your projects.

Capers Jones' article, *Software Cost Estimation in 2002*, provides a good overview of the types of functionality that are typically present in the approximately 50 commercial software estimation tools marketed in the United States. He covers 10 generic features most often found in many of these tools and conjectures about future trends in software estimation. This article is a good tutorial on the basics of software estimation. Alex Lubashevsky's article, *Early Estimation of Software Reliability in Large Telecom Systems*, describes how using two estimation techniques simultaneously can provide successful results. He used a U.S. Air Force Rome Laboratory model for an early estimation of software reliability, along with a second estimating technique derived from company-wide software process assessments based on the Software Capability Maturity Model® (SW-CMM®).

The revolution of new business strategies employing Web technology has highlighted the need for new ways to measure development time associated with Web projects. In his article, *Estimating Web Development Costs: There Are Differences*, Donald J. Reifer describes a new size metric, Web objects, and a new cost estimation model, WEBMO, that have been developed to satisfy the estimation needs for Web projects. He also discusses some of the key differences that managers must recognize between traditional software development and Web development to be successful in accurately estimating their projects.

The article by David Garmus and David Herron, *Estimating Software Earlier and More Accurately*, is a good synopsis of the function point method of software cost estimation. It claims that utilization of a functional sizing technique such as function points provides the capability to accurately estimate a project early in the development process.

Although not specific to software cost estimation, the article by Dr. Greg D. Power of Sverdrup Technology, Inc., *New Code Analyzes Fluid Flow for Better Designed Aerospace Vehicles and Components*, demonstrates a new way of doing software development for government aerospace applications involving analysis of fluid flow. This article reports on a partnership between the Air Force's Arnold Engineering Development Center and the NASA Glenn Research Center for development of a computational flow simulator.

Nancy Redgate and Dr. Charles Tichenor identify in their article, *Measuring Calculus Integration Formulas Using Function Point Analysis*, specific steps for handling mathematical formulas using function point analysis techniques. Such integration formulas are typically embedded in many engineering and scientific applications. The methodology does not require any new counting rules or patches and promises to give users a more accurate view of application size, resulting in better forecasting of costs, schedule, and quality.

Finally, David Henry gives us some practical lessons learned in his article *Software Estimation: Perfect Practice Makes Perfect*. He focuses on utilizing input from all the engineers involved in a software development project in order to develop personal software estimation accuracy, and on using actual historical performance data to assist in future estimation efforts. He also covers simple methods for group estimation techniques that he has found effective for creation of an organizational estimation process.

I hope that our theme articles will provide a little more insight into your challenges in developing an accurate software estimation capability within your organization. You may want to consider using CROSSTALK to report your organization's successes and lessons learned in cost estimation or any other unique software development capability progress.

Deputy Director, Computer Resources Support Improvement Program