

IT METRICS STRATEGIES

Helping Management Measure Software and Processes and their Business Value



! " ! # \$
% & ' ()

In recent issues of *ITMS*, we looked at managing the size of IT applications. This is critical whether you're building IT applications inhouse or contracting an outsourcer or IT partner for applications development and maintenance. In both cases, mismanaging the scope and size of commitments can have dire results. Overpromising can result in the project overruns and slippages that are at epidemic levels in our industry.

Two common metrics for software size are source lines of code and function points. But by no means are these the only units of size; others include objects, modules, programs, components, and frames. Obviously there are scaling relationships between these abstractions. How can you translate one from the other to understand the proportional aspects of one metric to the next? If you knew the size of a major application in source lines of code, is there a way to equate the size using a metric like function points for people who speak that language?

For example, if you knew the square footage of a house, and the amount of building materials used, could you approximate the number of rooms (if you knew the scaling relationship between types of rooms and typical square feet per room for that design)?

Addressing this topic are Carol Dekkers and Ian Gunter. Their article tackles the subject of deriving function points from code, which is known as backfiring. Why is backfiring used by some organizations? In addition to the examples cited above, some advocates of function points desire a "common currency" of size, irrespective of the code size, which can vary due to different software development languages. But backfiring has its problems, and you should read this article to understand some of the issues you'll need to consider.

Also included in this issue is a review of an important new IT book, *IT Metrics: A Practical Approach*, by Anita Carleton and William Florac. Jim Heires, a front-lines practitioner of metrics, benchmarking, and process improvement, takes an in-depth look at the advice and knowledge contained in this work.

Michael Mah, Editor

Using "Backfiring" to Accurately Size Software: More Wishful Thinking Than Science? 1
 Determining Your Own Function Point and Lines of Code Proportions — Three Things to Watch Out For. 1
 9



