In this special focus section, “Conceptual Association of Functional Size Measurement Methods,” by Onur Demirors and Cigdem Gencel, addresses an aspect of a concern typically found in business data processing: how to measure the business value of the software to be constructed. The problem is unresolved; we still don’t know how to measure the value of what we’re producing. So, we’ve created a number of surrogates for business value, such as lines of code.

One early approach counted the things that are part of the problem’s solution, are invariant to the method (for example, lifecycle model) used, and don’t depend on the actual count of instructions. This approach didn’t depend on the specific implementation details, such as the programming language. The intuition was that we can derive this count when first constructing the solution and that the count then stays the same throughout development. So, in some sense that count characterizes—or at least is related to—the value of what will be delivered.

Part of the struggle to find a good measure of business value is that it has fostered something of a cottage industry of alternatives. One of them, functional size measurement—function points—has several competing variants. Demirors and Gencel find that three standards for counting the size of “functionality” have much in common. They seek a way to treat these standards as specializations of a unified, generic model. They share with us this model’s real-life application on a project on which they worked.

Speaking of business, Magne Jørgensen presents “How to Avoid Selecting Bids Based on Overoptimistic Cost Estimates.” His article is valuable in many respects, most notably because it offers advice to both sides of the software acquisition equation: acquirers and bidders. It’s essentially a set of heuristics or rules of thumb that have an empirical justification and that you can use to establish the acquisition’s structure and to guide the participants’ behavior.

Of particular value is that Jørgensen draws from a corpus not usually familiar to us in the software development and acquisition community: the economics of competitive pricing and bidding. One noteworthy topic he describes briefly, of which it would be valuable for us to be more aware, is the “winner’s curse”: that is, to win only when bidding overoptimistically. We’ve observed a lack of discussion of this topic in software engineering and project management textbooks, which suggests that there’s little awareness of how the winner’s curse affects cost overruns.

Besides the benefits of reading these articles that I’ve already articulated, there’s one other: they’re created especially for the IEEE Software audience, particularly software project managers. Each article relies on longer, more detailed articles published by the authors elsewhere. So, if your interest is piqued, you can read more about the subjects by beginning with the articles’ reference lists. In other words, these articles take the larger, multifaceted topics and aim the results at you, the reader.

In sum, both articles give us a chance to dive deeper into measurement and understand its nuances and consequences.

Stan Rifkin, Master Systems

Measurement is a standard practice in engineering and in management, too. We believe that measurement increases the depth of our understanding, and we express a normal impulse to try to measure what we do or seek to do.

About the Author

Stan Rifkin is a founder and principal of Master Systems, an advisory service for organizations for which computing is strategic. His research interests are how to accelerate the adoption of practices at the organizational level—that is, how to accelerate improvement—and what is the best way to perform software management functions. Rifkin has a doctorate from George Washington University. Contact him at sr@master-systems.com.