

## Learning to Measure or Measuring to Learn?

Anneliese K. Amschler Andrews  
*Department of Computer Science*  
*Colorado State University*  
*aaa@cs.colostate.edu*

Since the First International Software Metrics Symposium in May 1993 in Baltimore, our community has come a long way. In the beginning, our discipline concerned itself with measurement theory, how to define meaningful measures that are credible, valid, and reliable. Much effort was expended in defining what one should and must not do with certain types of measures.

Industrial efforts developed metrics programs. Key issues centred around criteria for success and how to predict the success or failure of a measurement program. In the wake of increased metrics activities, the number of metrics proliferated. Over time, so did the analysis methods. As a result, we now have a large arsenal of quality metrics and a host of statistical methods.

Increasingly, however, the focus has shifted away from defining ever more metrics and subjecting measurement results to increasingly sophisticated statistical analyses to an emphasis of understanding software phenomena. This has led to more case studies, empirical work like field studies, as well as experiments. This emphasis can be clearly seen in this year's METRICS program. With this, the methodological concern shifts from measurement theory to proper design of studies. It also has caused a new look at use (and abuse) of various statistical analysis methods.

With all our metrics definition and measurement program efforts, much of the data available to us is still limited, incomplete, or subjective. If we want to learn from such data and use it to make good decisions, we need to assess software and software processes from multiple angles and with a great deal of flexibility. Our arsenal of tools is huge. To expect only a few to help us find answers to complex problems is about as realistic as using a hammer for every handyman job in the house. This keynote explores some of the ways our discipline has developed and how multiple tools can be used with qualitative and quantitative data to understand complex software phenomena better so as to make better decisions. We illustrate this on a study of a testing organization that combines a variety of metrics, models, and analysis techniques to understand strengths and limitations of the software under test and the testing process.