Engineering software engineering metrics

Robert M. Poston, Software Standards Editor

Something is wrong with the software metrics we have now. Few professionals are systematically measuring the quality of their software products or the cost and schedule of their development activities.

Yet everybody I know who works in software development performs some task that involves estimating (predicting or forecasting) product quality or process cost or schedule. Not only do these people have to make estimates, they must evaluate products or processes. They also may have to select from alternative products or processes.

Estimating, evaluating, and selecting are routine duties of top-level executive managers, middle-level project managers, and members of the technical staff. The scope of responsibility may differ, but these same fundamental activities must be performed at every organizational level.

Trained as we are in math and statistics, we know we can make better estimates, evaluations, and selections if we have quantifiable information to work from. The importance of software measurements is understood. The need for them is widespread. So, why are today’s software metrics being ignored? Very simply, they don’t satisfy user needs.

That’s a problem we are all too familiar with in software development. Barry Boehm tells us that historically we’ve been weak in setting requirements for software products. We have trouble understanding what the user really needs (often because we don’t know who the user is). Of course, when we don’t fully understand what the user needs, we run into great difficulty trying to specify requirements for something that will fulfill those needs.

Maybe it is time we engineered an integrated set of software metrics. When we develop a software product for a customer, we follow a defined development methodology (at least, that’s what most companies’ marketing departments would have us believe). Since our metrics don’t seem up to snuff, why don’t we use a methodological approach to create some really serviceable metrics for ourselves?

Let’s make a stab at defining some requirements for new metrics right now. If we were to undertake a professional project to define requirements for new metrics, we would divide our work into the three tasks of analyzing user needs and problems, specifying requirements for the product, and verifying that the requirements are correct, consistent, and complete.

Identifying metrics users and their needs. Who is the metrics user? If we are to believe previous metrics developers, the only user is the project manager. We see now that this is not true. In fact, everybody in an organizational chain must use metrics to be effective in estimating, evaluating, and selecting.

Members of the technical staff must estimate how long it will take to produce a product (document, program, or test). They often are asked to examine the functionality and size of a program so they can predict how long it will take to execute that product. The technical staff member is a “now” user typically needing metrics to make estimates that affect a current activity or one right around the corner.

In middle management, there are two heavy users of metrics: the project manager and the functional manager (one who controls skill pools made up of such people as systems engineers and programmers). The project manager’s need for metrics is the most obvious. Schedule and cost-estimating for the development process is a way of life for this person.

Most of the time, project managers are not required to estimate product quality (few managers at any level realize how critical product quality is to driving cost and schedule factors). The project manager is a near-term user estimating, evaluating, and selecting processes and products just for the duration of a project. Estimates and measurements produced by the technical staff should be used by the project manager.

The functional manager loans or contracts people to projects keeping two criteria in mind: knowledge of a technical area and skill or experience level. This manager uses metrics for near- and long-term projections. How many and what kinds of experts will be required to complete present projects? What about staffing future projects? The functional manager will use the estimates and metrics compiled by the technical staff and project managers as a foundation for his projections.

The executive manager of a company that produces software must do very long-range forecasting. Executive managers are usually found in one of two situations.

In the first situation, the executive manager pursues business opportunities for the company to build a product for a user. This often is called special projects business. Most military contracting falls into this category. Here, the executive manager is under pressure to select, from many requests for proposals, the few that are most worth investing in. Without a history of quantitative measurements on previous processes and products, the executive manager will be shooting in the dark.

In the second situation, the executive manager must decide what products to build and sell to mass markets. In addition to a history of quantitative measurements of internal processes and products, this executive will want information about the operations and products of competitors.

We can see that metrics must address the needs of whole teams of users. Trying to focus on the needs of just one user has led to metrics most of us don’t use.

We have seen that people at different levels of an organization have different needs for metrics. Analyzing user needs becomes even more complicated when we realize that people at the same level may wear different hats or play different roles.
Take, for example, the technical person who is defining a product, controlling changes to it and testing it. In each of those capacities, the technical staffer needs different measurements to make reliable estimates.

What about the project manager? Sometimes he must wear the quality assurance hat. At other times the configuration management or test management hat is required. We begin to understand more of the complexities.

About 15 years ago, I was involved in a metrics collection effort for a large aerospace company. My primary job as a member of the technical staff was the creation of real-time radar process-control software. A manager came through our shop one day and said he wanted us to fill out forms to collect metric data. The information was to be used by executive managers to forecast future work.

I was working 12 to 14 hours a day, trying to meet a poorly forecasted schedule. The forms seemed like an added pain in the neck. Every so often I would hurryly scratch out what information I could remember so I would have something on the forms. Certainly the information was not well-defined and was not going to be of much value to executive managers. I couldn’t see that the metrics collection was helping anyone.

People collecting metrics need to know how this work is benefiting them. Only then will they take this task seriously. And, of course, reliable forecasting, evaluating, and selecting can be done only when accurate, complete information is provided.

Specifying some requirements. Granted, this has been a quick pass at identifying metrics users and some of their needs and problems. But, already, it has let us pinpoint a significant requirement.

Each metric must be a proper subset of an integrated set of metrics. The integrated set must be usable by every member of the team as he performs various jobs. A stand-alone metric that will satisfy one need for one user in one situation is not going to help much.

Having an integrated set of metrics would lower the cost of data collection by distributing the work throughout an organization. Each person in the chain would gather and use metrics information about his own job and then store it in a database. Managers could access information from those down the chain. Naturally, the more automated the collection process, the better. But, that is a subject for another day.

Since the database is going to be used

---

**ENGINEERING PROFESSIONALS**

We're the Government and Defense Group of Emerson Electric Co., a highly diversified firm. We are looking for a select group of professionals to work on several long term contracts.

If you are interested in the following positions, we would like to hear from you.

**ATE ENGINEERS**
- Project Engineers and Staff Engineers to plan, estimate, direct, monitor, and control all Engineering hardware and development tasks of a given ATE project/program from proposal preparation through system development. Requires minimum of 8 years experience in design and development and/or engineering management of test equipment systems with emphasis in RF & Digital.
- Design of Test Systems hardware and Test Program Sets. Requires minimum of 3 to 10 years experience in electronic systems testing and design including programming experience in ATLAS, FORTRAN, Pascal, Ada, JOVIAL or Assembly languages.

**SOFTWARE DESIGN**
- Real-time signal processing software development experience in 6809, 68000, and 68020 micro processors. Additional positions require experience in the design and development of Test Systems Software and application of systems. All positions require experience in one or more of the following systems or languages: VAX/VMS, 1750A, FORTRAN "C," Ada, JOVIAL, FORTH, Assembly languages, and ATLAS.

All positions require a minimum of 3 to 10 years experience with BSEE or BSCS/Engineering; MS preferred.

**RELIABILITY/MAINTAINABILITY**
- Perform reliability and maintainability related analysis of electronic and mechanical systems. Prepare proposal and program plan inputs. Interface with design, components, failure analysis, logistics, and test engineering personnel.

Requires minimum of 2 to 7 years electronic circuit analysis and preparation of reliability related analysis. Must be familiar with reliability and maintainability specs and program requirements.

All positions require a BSEE, BSEET or BSCS degree. Positions are in St. Louis. We also have select ATE positions in our Orlando Engineering Center.

To arrange a personal interview in your local area, please call TOLL-FREE 1-800-325-0783 or send resume to:

Irma Shillito, Sta. 4304
Electronics & Space Division
EMERSON ELECTRIC CO.
8100 W. Florissant
St. Louis, MO 63136

---

**EMERSON ELECTRIC CO.**

Government & Defense Group

An equal opportunity employer.
by people at different levels in the organization, the metrics information must meet another requirement. It must be normalizable. Consider, for example, that, on one task on a project, a programmer may be using assembly language while, on another task on the same project, a programmer may be working in Ada.

How can metrics information collected from those two tasks be normalized so a process schedule can be estimated? How can an executive manager, predicting cost and schedule for a future project that will use fourth-generation languages and prototyping, apply database information derived from projects that did not use such modern elements?

These are the kinds of factors that would have to be considered in defining requirements for an integrated set of metrics.

Accuracy in metrics collection cannot be stressed too much. Once collected, the data is usually used in models such as Barry Boehm's COCOMO or John Musa's reliability predictor.

The models are sets of equations that employ the metrics as parameters or coefficients. Any errors in the metrics will be amplified in the use of an equation. For example, if we multiply two numbers, both having error margins of ±20 percent, the result will have an error margin of ±40 percent.

This simple error propagation illustration shows up another requirement: Software metrics must be accurate within some allowed tolerance. I would like to see that tolerance set no higher than 10 percent.

This is a very restrictive — but necessary — requirement. It says all software metrics will be explicit, simple, directly obtained indicators such as number of inputs and outputs to software products or number of repeatable test cases.

This eliminates as possible metrics for our use analytical or complex measures that are based on assumptions about the relationships among software features (for example, Halstead length and cyclo-metric complexity). The assumptions are subject to too many errors.

It also eliminates subjective ratings of quality that are arrived at by high-level review, inspections, or comparisons (such as traceability and completeness). The error margins of subjective ratings have proven very high.

You may ask, "What does this leave us?" We do have a few primitive software metrics that can be counted repeatedly from project to project: number of inputs and outputs (primitives and structures), number of fixes applied to code (in contrast to errors, faults, and failures), and number of separately testable collections of code. These are good starting points.

Four standards projects under way.

Four IEEE standards efforts aimed at defining metrics are now under way. Reliability, productivity, quality, and error/fault failure classifications are the focuses of these working groups. So far, these people have only scratched the surface of their complex areas. A lot of work remains to be done.

What requirements have these groups set for the metrics which their standards address? How did they identify metrics users? Which existing metrics have come closest to meeting their requirements for standardization? Interesting discussions lie ahead.

Please contact the chairman of any metrics working group (see the accompanying box) if you have ideas or time to contribute. Write or call me if you have any comments or positions you would like to discuss in this forum. It's important we keep our lines of communication open as we continue to tackle our metrics problems.

### Working groups

**P982 — Standard for Reliability Measurements**

James Dobbins  
IBM FSD  
9500 Godwin Dr.  
Manassas, VA 22110  
(703) 367-3912

**P1044 — Standard for Classification of Software Errors, Faults and Failures**

Richard Evans  
TRW MS 144-1065  
1 Space Park  
Redondo Beach, CA 90728  
(213) 535-7791

**P1045 — Standard for Software Productivity Metrics**

Robert Sulgrove  
NCR  
MS WHQ-SE  
1700 S. Patterson Blvd.  
Dayton, OH 45479  
(513) 445-1064

**P1061 — Standard for Software Quality Metrics**

Norman Schneidewind  
Naval Postgraduate School  
Dept. RSA/CS, Code 548S  
Monterey, CA 93940  
(408) 646-2719

---

### Standards bulletin board

#### Program abstract review

The public review and comment period for the draft of the proposed reaffirmation of ANSI X3.88-1981 ends Oct. 1. The standard defines the content of computer program abstracts, which summarizes the capabilities, operating environment, and other descriptive information.

Copies are available from the American National Standards Institute, Sales Dept., 1430 Broadway, New York, NY 10018; (212) 642-4900 at $8 per copy (including postage).

#### Posix status report

The June meeting on the Posix trial use document (P1003.1 working group) resulted in several recommended rewordings, mostly for clarification. The proposal on job control concepts was reviewed and will be proposed this month as an optional part of the next Posix draft.

One common interest was the relationship of the Posix standard to the System V verification tests developed by AT&T. Appendix I generally describes the differences, but most reflect Posix extensions not found in the current version of Unix.

Two new working groups have been spun off from the 1003.1 effort: the 1003.2 group for shells and tools and the 1003.3 group for verification test specifications.

For more information, contact Jim Isaak, Charles River Data Systems, 983 Concord St., Framingham, MA 01701.

#### Software engineering

The next meeting of the Software Engineering Standards Subcommittee is Nov. 4, and will be held in Dallas at the Fall Joint Computer Conference.

Balloting groups are now being formed for four new proposed standards. The standards are P1028 Software Reviews and Audits, P1042 Software Configuration Management, P1058 Software Project Management Plans, and P1063 User Documentation.

For more information, contact John Horsch, Teledyne Brown Engineering, 300 Sparkman Dr., Huntsville, AL 35807.