SEPM: Challenge refused?

Editor:

In one of the special feature articles of the August 1980 issue, Thayer, Pyster, and Wood identify 20 "crucial problems" in SEPM and issue a challenge (on page 52) for refutation of any one of these problems (which are listed on page 53 of the issue). The challenge is to refute "by preparing and publishing a technical paper...for publication in...Computer or IEEE Transactions on Software Engineering."

With imperceptible success, I have for some time attempted to stunt the growth of software engineering, the father (or parent) of SEPM. Accordingly, I am unwilling to respond to the challenge to write a technical paper which could come under the SEPM umbrella and which in some way would plump up or give legitimacy to SEPM. (I realize the futility of this effort, and I am steeling myself to the occasion when I meet my first Bachelor, Master, or Doctor of SEPM.)

The 16th problem listed in the enumeration of crucial problems is the only one I will discuss and the only one I feel qualified to discuss. This reliability problem, as stated, is: "Measurements or indexes of reliability that can be used as an element of software design are not available, and there is no way to predict software failure, i.e., there is no practical way to show the delivered software meets a given reliability criteria." There is a good thought embedded in the primary clause of the statement but it is polluted by a very bad (i.e., untrue) statement in the subordinating clauses.

First, the good thought. It is now true, and it may always be true, that, except for a trivial direct and obvious relationship between the number of errors and the instruction count, there is no set of measures (Halstead's metrics notwithstanding) which can employ design parameters to produce an estimate of software reliability of a delivered package. Thus, the initial clause appears to be true—but only in isolation.

The bad thought or statement is that "There is no way to predict software failure." This statement does not tie to the former "good thought" just discussed. No one can ever project from gross measurements of code and of associated flowcharts or directed graphs, the time of the occurrence of a failure, for the latter, as measured by mean-time-to-(next)-failure, is a changing metric which does not explicitly depend on the gross quantitative parameters of the code, but rather depends on the amount of testing which has been done on the package. (One can derive gross estimates of the error content of a delivered package, but this is of little use in practical situations—no one ever makes an initial delivery of a complete and untested package.)

There are many practical ways of developing reliability estimates. J. D. Musa of Bell Telephone Labs routinely estimates the (continually changing) reliability of software packages of substantial size. I have applied several models to about seven projects with documented success. Goel and Okumoto have produced exceptionally accurate measurements of reliability. Motley and Brooks have recently developed a model which can be used during a program's development phase. Prompted by their success, I modified the stepwise rate model which I and Z. Jelinski had developed, so as to permit MTTF estimates during the very early stages of software development.

All of these practitioners have had good success when they applied their own models to data. Some others, who do not exercise requisite caution as to the regime of applicability, have had little or no success and have "blamed" the models.

I would welcome the opportunity of carrying on further discussions in "The Open Channel" section of Computer.

Paul B. Moranda
McDonnell Douglas
Huntington Beach, California


Challenge renewed

Editor:

We feel a little like the messenger who was killed because he carried bad news. Mr. Moranda's sensitivity to software reliability is understandable since he has written several articles on the subject. However, our challenge is not based
just on our personal views on the 20 issues. We have only acted as reporters. Each issue was a consensus of hundreds of project managers, programmers, researchers, and others from the computing field.

Since Mr. Moranda is unwilling to describe his proposed solution to problem 16 in a form that can be used by project managers (software or otherwise), we must simply count his letter as a vote against that issue being considered as an important problem. This revises the total votes regarding whether or not the proposition is an important problem from 249 "aye" and 45 "nay" to 249 "aye" and 46 "nay."

Managers of software engineering projects need both technical and procedural methods by which they can deliver software on time and within cost that is usable to the customer. Toward this end, a special issue of the IEEE Transactions on Software Engineering has been reserved for early 1982 for a set of practical papers on software engineering project management. So, the challenge still stands—despite Mr. Moranda's reservations.

If any reader has a practical, down-to-earth solution to one of the many problems besetting a project manager of an average size project (5-10 people, 1-2 years) that can be implemented by a manager in an organization without unusual resources, we would like to hear from you. Better yet, document your solution with a paper, so it can be used by others.

Richard H. Thayer
Arthur Pyster
Roger C. Wood

Baerly punny

Editor:

One tends to doubt Selim Akl’s notion (Computer, October 1980) that using the name “Descartes” for the next version of Fortran might “win the hearts of our European colleagues,” from whom we hear that Fortran is “doomed.” They would merely consider this to be placing Descartes before the hearse.

Robert M. Baer
Metadata Corporation
Mill Valley, California

Ballot design

Editor:

In the interests of human engineering and so we won’t appear to be a complete set of nits, I wish to suggest that for our next Computer Society election we print the “punch-out” boxes as the leftmost column on the ballot, followed closely by the names of the candidates. This would eliminate the myriad problems in alignment and registration experienced in the recent election by those like myself with minimal motor skills and visual acuity.

Then, perhaps, we could go on to address some of those “big picture” problems outlined by the candidates in their statements...

John Seitz
Informabile, Engrg.
Quebec, Canada

To be considered for publication, a letter to the editor must be accompanied by a statement giving Computer permission to publish that letter.