Copyright Notice

To H.D.Mills, Chairman Software Methodology Panel.

Dear Harlan,

I am not quite sure how to comment on "Essential Elements of Software Engineering Education" by Peter Freeman, Anthony Wasserman and Richard E.Fairley, because I don't like its underlying political assumptions, because I know that, when dealing with politically distasteful attitudes, my pen tends to get venomous, and, finally, because I don't particularly want to offend anybody. So I hesitate.

There is, for instance, the authors' view on the proper role of our universities. They include producing the graduates industry and government ask for. An alternative view is trying to educate the graduates the rest of the world will need in the future, independent of the question to what extent the rest of the world already understands its future needs. This may sound presumptuous, but universities are by definition --if they are any good-- presumptuous institutions with targets more far away in the future than most other organizations. I definitely prefer the alternative view, for where, otherwise, is the necessary innovation to take place? The degeneration of our universities into graduate factories is a development I would not like to encourage, because I consider it to be a threat to our civilization.

There is, for instance, the authors's view on the role of the intellectual individual. With their stress on the supposed virtues of group activity (and on the need for "communication skills"!) they seem to regard minimization --or possibly even elimination-- of his role as an ideal worth to be pursues. I regard that as a threat to our civilization. (For further details I refer you, for instance, to "The Organization Man" by William H.Whyten first published by Simon and Schuster, New York, 1956)

There is furthermore the observation that of their Five Pillars of Wisdom for the software engineer --computer science, management science, communcation skills, problem solving and design methodology-- only the first is hard science, while the remaining four --if existing at all-- range from soft to very soft. I am afraid, however, that the current fashion grossly overestimates the importance and potentialities of the soft sciences, and would not like to enforce that fashion, because, again, I regard it as a threat to our civilization. (For further details I refer you to "Social Sciences as Sorcery" by Stanislav Andreski, Penguin Books, 197?.)

Finally --as most political documents-- it is superficial. The suggested analogy between the software engineer and the family doctor is false, because the commitment of the medical profession and the commitment of any engineering profession are of quite different natures. The most blatant example of superficiality is probably their argument in favour of communication skills, as they refer to "the software engineer's need to communicate with a wide range of people and machines". As a piece of hilarious nonsense I think that this is only surpassed by "The education of a computer" (Grace Murray Hopper, Proc. ACM National Conference 1(Pittsburgh, 1952) 243 - 250).

So, if you intend to follow Raymond T.Yeh's suggestion to use the paper by Freeman et al. "as a basis for departure", I can only recommend that you depart from it as far as possible.

* * *

C.V.Ramamoorthy's "Preliminary Report on Software Evaluation" is less objectionable: it gives a survey of what is or has been done --no matter how sensible or how foolish-- and I have not the slightest reason to assume that his survey is unfair or incomplete. The report is very instructive, even perhaps in unintended ways. This does not imply, that I have no objections: the author fails to challenge the assumption that the whole approach makes any sense at all. Let me quote:

> "The approach is to identify a set of software characteristic attributes representing good and bad, reliable and unreliable programming practices. For each attribute, measures called metrics are formulated. The merit figure of a program is then defined as the normalized weighted average of these attribute metrics. The validity of this approach depends heavily on the chosen attributes, the metric formulation and the function that combines these metrics."

The first sentence is OK, but in the second sentence the word "measure" is used in a most unscientific sense. In science we measure physical quantities, something that is a meaningful activity because (the measurements of) these quantities are supposed to satisfy certain explicitly stated laws; the purpose of the measurements is to confirm or to refute the supposed laws. Here, however, to "measure" is used in the sens of "attaching a number to", in very much the same way as psychologists construct an IQ. (It is a fallacy to assume that an IQ "measures" something!) The next sentence is OK in the sense that it describes a common practice, be it a deplorable one, the last sentence is wrong in that it assumes that the notion of

"validity" is applicable to such practices, in that it assumes that some of these practices can be more "valid" than others: validity is a binary criterion.

In his research (!) recommendation the author shows --apparently without noticing it-- that the problem is recursively unsolvable: here he suggests that the criteria used in evaluating the quality of software, in turn should be evaluated themselves for their effectiveness. (And so ad infinitum...!) The recommendation ends with "These effectiveness measures may allow a user to include an optimal set of tools in a software evaluation system to meet his special needs." No matter how hard I tried, I could not attach a sensible meaning to that sentence; presumably it will be discovered that more "research" should be devoted to the quantification of the user's "special needs", so that we can decéide whether a set of tools is "optimal"!

The whole activity has very little to do with what I would like to regard as software engineering. It is more the further refinement of management "science" as a self-perpetuating activity. If the manager needs a number, he will get one. I am afraid that the whole activity is adequately captured by the well-known saying "If you ask a foolish question, you will get a foolish answer.".

Note (added to avoid misunderstanding). I don't know how to manage the design (or should we say "the discovery"?) of software. I hope that my saving grace is that I don't pretend to. (End of note.)

Good luck! Yours ever,

Edsger W. Dijkstra

Plataanstraat 5
NL-4565 NUENEN
The Netherlands

prof.dr.Edsger W.Dijkstra
Burroughs Research Fellow

PS. May I ask you to distribute this text to the other panel members? I don't have all their addresses, and, besides that, most of them are located in the USA. Thank you.

EWD.