Letter on Parnas' View of Dijkstra vs. deMillo, Lipton and Perlis

H. Kilov

I disagree with Dr. Parnas' view in the October 1978 SEN (3 4, pp. 20) that there are two kinds of programs -- useful ones (the result of software engineering) and, let us say, textbook ones (intended to be understood by someone else). That the latter may be useful is shown by Dijkstra (updating a sequential file, string pattern matching, ...) and others. Of course, often good textbook programs seem too simplified -- maybe, because we are used to the need to adapt to and sometimes to fight with the existing software systems. But then the software systems are to blame!

I think the correct analogy is not between software engineering and mechanical engineering, but rather between software engineering and applied mathematics. (The essence of the latter is very well described in the book "Applied Mathematics: Subject, Method, Approaches" by Blekhman, Myshkis, and Panovko, published [in Russian] in Kiev in 1976.)

Finally, I think sometimes it is quite possible to "test" theorems in mathematics -- i.e., to find counterexamples to them. (In fact, journal mathematics proofs can contain errors as well!)

Karl Marx Str. 75-13
Riga 11 USSR

H. Kilov

An Alternative to Structured Programming: Syndicate Programming

Recent work [1,2,3] has pointed out the need for the organization of programming teams and for efforts to improve software quality and reliability.

The idea of Chief Programmer Teams, although well thought out, tries to induce an artificial and unproven organizational structure in the programming effort. It seems, however, that a structure should indeed be used to curb the amount of poor quality being produced nowadays. Careful sociological and anthropological studies show that two social groups have reached exceptional organizational efficiency, the Papuan tribes of New Guinea [4] and East-coast organized underground illegal organizations [5]. In this paper we examine only the second, as the high efficiency of the first seems to be somehow related to the low demands of their prehistoric life -- a situation not at all similar to that of the world of software (except for the use of prehistoric tools).