

My comments fall in three distinct classes.

1. Is an international institute the proper vehicle?

The report of the NATO Study Group on Computer Science is not very convincing on this point. If the goal is so important it fails to show why private enterprise could not undertake such a project. The fate of co-operative bodies such as ICC, in Rome is a warning that should not be ignored. Particularly in this case, where the competence of the top-people engaged is of so vital importance, the usual boundary conditions on nationality can only be expected to be very harmful.

2. How has the project been chosen?

Having heard Dr. M.D. McIlroy's talk at the NATO Conference on Software Engineering at Garmisch, October 1968 and having read the aforementioned report, one cannot escape the impression that a very tentative suggestion made by Dr. M.D. McIlroy has been welcomed as a means of giving substance to an otherwise empty construction. Some of the considerations in the report strike me less as motivations having led to the worthiness of the cause than as afterwards dressing it up in OK-sentences. I quote "An area ripe for engineering progress is the fabrication, along standardized lines, of a range of software components that can efficiently bridge the hardware-software gap in a multiplicity of actual applications on a multiplicity of actual machines."

3. How feasible and how desirable seems the project?

I would classify the talk of Dr. McIlroy (for whom I have the greatest respect) as a worthy contribution to the famous and much-needed "Journal for half-baked Ideas."

It has been inspired and recommended by an analogy to the automobile industry, as is still reflected by the sentence in the report: "This would be an attempt to bring about a change from handicraft to industrialized production with consequent improved efficiency and reduction in price" but the analogy is misleading for in contrast to any hardware industry, the cost of making replicas of a design is small in the software industry. I fully agree that software engineering should rise as soon as possible above the handicraft level (it does already!) but to tackle this by "automated software [component] production" before the basic disciplines of programming are much better understood than they are now strikes me as rather premature.

Even if it could be done I have grave doubts as to its desirability. Underlying the stress on "component production" we find the tacid assumption that, macroscopically speaking, present-day systems are desirable in structure and that our sole difficulty to-day lies in our insufficient ability to make such systems. I would venture the opinion that the main source of our present-day troubles is not our production technique but the kind of object we are trying to produce. Standardized component production, once established, seems hardly a fertile ground for a more basic rearrangement of our thoughts about system structure. And it is exactly there that I expect the greatest potential benefit.

Edsger W. Dijkstra.

Edsger W. Dijkstra.