Dear Editor:

This week the shortage of software talent has made the headlines, as the government recognizes the economic importance of the systems these folks build and the consequences of not having enough people to build them. It is not the reported 340,000 current job openings that is the problem, however. It's the last hundred thousand software developers who got hired as firms became increasingly desperate to fill open positions on software projects of all sorts. One of those new hires may be debugging the air bag system in your next car. Another may be testing the nation's next-generation air traffic control system.

Current proposals to address the nation's shortage of IT talent include measures to transition people from other fields, retrain people with obsolete skills, and even train the hard-core unemployed to do programming. In Malaysia, where the shortage of talent is recognized as a major barrier to that country's high-tech future, programs have been proposed to "reskill" ex-convicts and drug addicts.

Once upon a time, it seemed that anybody could be a programmer. In the 60's, for example, when the first wave of business and government computing created the need, hundreds of thousands of engineers, scientists, accountants and clerks found their way into this new profession. They did a great job of creating the software that is now running our national enterprise. Over the years, however, the functionality and complexity of software has increased dramatically. Simultaneously, our businesses and our lives have become dependent on millions of programs working correctly almost all the time. Software is serious, professional work — well paid work — but software professionals are not licensed, often not formally trained beyond the "language" level of software engineering, and still not respected for the value they've collectively created.

Furthermore, most people who want to do software for a living are already doing it. Like any other profession that is dependent on specific aptitudes and skills, the best software people are a lot better than the average practitioners, who in turn, are a lot better than the worst. In fact, most software managers will tell you that the worst people on a project often make a negative net contribution.

So now that the world's demand for software has outstripped the supply of talented people to design, develop, test and maintain these systems, who is getting hired? As the most talented migrate to jobs at companies already aware of the strategic importance of their software, like Microsoft, Motorola and Citibank, who's replacing them in the hundreds of thousands of corporate application development groups and government information systems departments around the world? And what will the consequences be for the economy and for our lives?

Avron Barr and Shirley Tessler
Stanford Computer Industry Project
January 14, 1998
scip@cs.stanford.edu