SCIENCE NEWS®

The Weekly Newsmagazine of Science

Letters

The human component

As a computer professional, I have thought a lot and had many discussions about the "liability" of computers and especially software ("A Digital Matter of Life and Death," SN: 3/12/88, p.170). Much as the government is disparaged these days for over-regulating industries, I think there are certain roles that only the federal government can fulfill. I am glad to see the government getting involved in medical-equipment monitoring, as tragedies such as that involving the Therac 25 machine need to be prevented.

It seems that the solution to problems with monitoring devices such as these, where there is no human intervention and the device deals directly with a patient, is to add some level of feedback. When the dosage of radiation emitted from the Therac machine increased 100-fold, there should have been some sort of feedback, self-test if you will, to monitor this level and make sure it was within acceptable tolerance limits.

Also, this concept *must* be extended to provide a level of accountability to the people responsible for overseeing machinery such as this. In my hypothetical case, if the self-test failed, a human operator could be notified before further dosages would be administered. But it is very important that the ultimate accountability be established with some human involved in the operation of the machine (with possibly the inclusion of the manufacturer as well, within limits).

I am excited that the FDA is pursuing a responsible course of action here, and hope they can find it in the governmental budget to pay appropriate salaries to qualified technical types so that the review process doesn't become a bureaucratic joke.

Courtney Prodehl
Oak Park, Ill.

The programming referred to in "A Digital Matter of Life and Death" is called real-time programming, which is not related to the more plentiful data-processing type of work. Some call it software engineering.

Most real-time jobs involve small- to medium-sized projects, and they are often done by one or a few people who work in an engineering environment and who are so unsupervised that their immediate superiors are computer-language-illiterate. These programmers, by the nature of their supervision, can very easily confuse testing with debugging.

Good supervisors in this profession are hard to come by because good programmers love what they do, are well paid for it and refuse to be promoted. In fact, the personality traits that make a good programmer are the same ones that make an inept supervisor. Another problem is that programming is a creative activity, which leads to an endless number of ways to do something and leads some people to jealously guard and defend their work.

Since profits are sooner or later dependent on software quality, a well-managed endeavor should force programmers to interact and openly share ideas and especially criticism. More experienced people should be made to take time out to read and criticize the work of less experienced people. Also, some serious thought should be given to forming a "testing" profession (please, not a four-year degree) so that testing will be taken out of the hands of the people responsible for getting the job done on time.

Larry Elden
Dearborn, Mich.