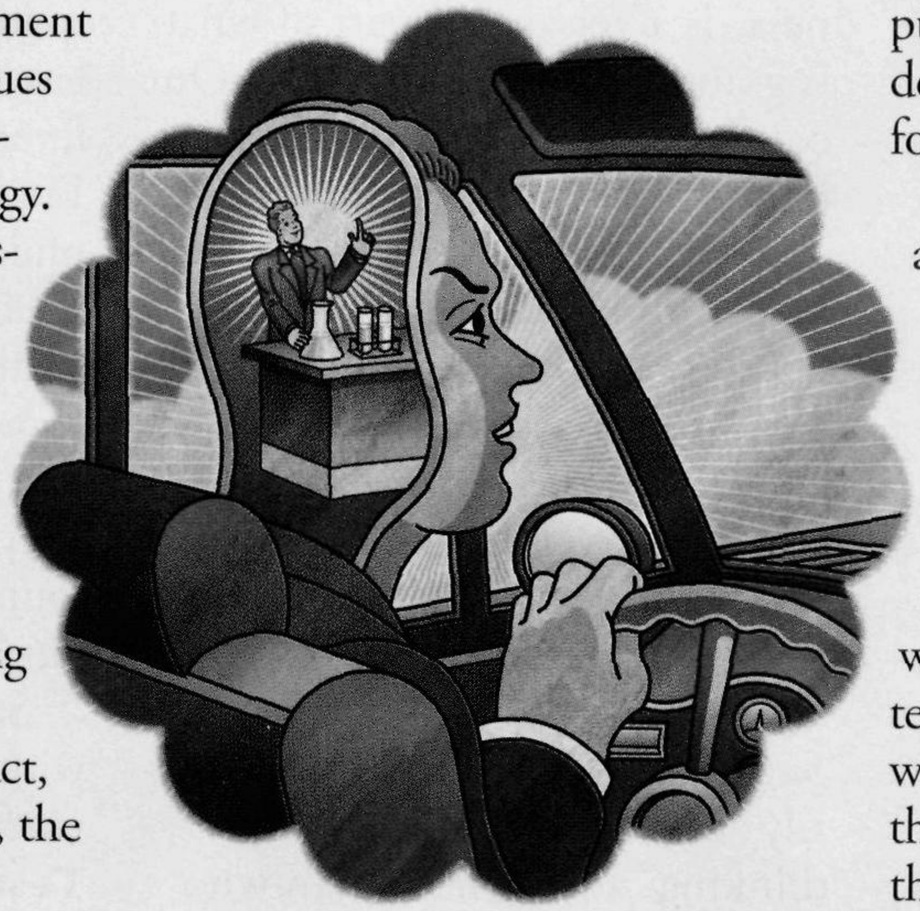
The Unconscious Art of Software Testing

The subtle psychology of testing.

In *The Art of Software Testing*, Glenford Myers asserted that "...the most important considerations in software testing are issues of economics and human psychology" [6]. In fact, the most important considerations of any software development practice are (or should be) issues of economics and human psychology. Particularly psychology.

The challenge in testing systems is that testers are trying to develop a way to find out if they don't know that they don't know something. This is equivalent to a group of scientists trying to devise an experiment to reveal something they are not looking for. It is extremely difficult to do. In fact, as Thomas Kuhn pointed out, the image we have of the scientist boldly going into uncharted territory and finding out things we never knew is at odds with the reality [5]. Scientists almost always find out things they already know. The hypothesis must come before the experiment to confirm (or deny) it. In fact, it is almost routine for scientists to ignore results that get in the way of their preconceived notions and carefully constructed intellectual models.

It is not possible to be wholly deterministic about testing since we don't know what to be deterministic about. Testing, probably more than any other activity in



PETER HOEY

software development, is about *discovery*. In the bad old days, people were sometimes punished for finding defects, since defects were considered bad. In my previous column, I pointed out that even the word "defect" is a little, well, defective [2]. By the time we get around to dynamic testing there may be things we should have found out earlier but didn't due to some negligence on our part. However, exposing things we didn't know we didn't know by dynamically executing the knowledge contained in a system is not itself a bad thing. We've stopped punishing testers for finding defects, though rewarding them for the same has its perils.

Sometimes the most effective and efficient way to find certain defects is to test for them. This does not in any way substitute for good engineering practices and feedback mechanisms such as inspections. Indeed, without such processes in place and working, any attempt at dynamic testing quickly becomes overwhelmed and quite ineffective. But there are certain kinds of problems that are very difficult to identify analytically. Some companies I work with perform enormous quantities of integration testing. They must do this because they have enormously integrated systems and it is very difficult to determine their behavior in operation unless you execute them in some controlled fashion. Simulation is taking over traditional testing in some of these areas, but a