What's Wrong?

My last vacuum cleaner failed in an obvious way: It broke in half. When I went shopping for a new one, the salesman demonstrated how well it was made by jumping on it. His testing procedure was convincing, and I bought it.

When it comes to engineering, I can think of two main reasons for testing:

1) Does it work?
2) Why doesn't it work?

Obviously these questions are related: We want to know *why* something doesn't work once we convince ourselves that it doesn't. The follow-on question "how do we fix it?" is often relatively easy once we know what's broken.

Engineers do a lot of testing. A lot. Engines and airframes are cycled innumerable times to find failure modes. Despite the best design procedures for every single component, testing is needed to confirm that the system works and is reliable. Testing might be an annoying expense, but engineers have been doing it for centuries, and they haven't stopped despite modern science and computation.

The systems that vendors deliver are supposed to work but don't always, for various reasons. Acceptance testing in these cases isn't a matter of distrust but just another level of quality control. OK, maybe a little mistrust.

Systems rarely work perfectly for the simple reason that few things are designed and built perfectly. But there are deeper reasons. First, components are often used in ways their designers never anticipated, and, second, complex objects and systems can possess modes of operation that were not originally envisioned. It's impossible to consider all situations that something will be subject to. Amazingly, some engineered systems are fairly robust, such as automobiles, which tolerate diverse driving habits and driving conditions.

Testing is even useful in mathematics, where proofs prevail. Sometimes proofs are difficult to understand and confirm with complete certainty. Subtle errors can persist for decades, long after experts have signed off on the correctness. But whether or not a proof is certifiably correct, I encourage students to "test" the math with a little skepticism. Subjecting a result to examples that satisfy the hypotheses of a theorem builds confidence in the result, whereas subjecting a result to examples that *don't* satisfy the hypotheses gives insight into its robustness.

We all know and use direct methods of fault diagnosis. If a system can operate without a component, then remove it and see if the problem goes away. If the component is essential, then substitute another component for the suspicious one and see what happens.

An indirect approach to diagnosing an anomaly is to simulate the system on a computer and play "what if." Components of the simulation can easily be altered, and the response can be compared to the faulty operation of the real system. This kind of heuristic detective work can be remarkably powerful.

Dennis Bernstein and *IEEE Control Systems Magazine* Associate Editor Rafael Fierro at the University of Michigan, where Rafael presented a seminar.

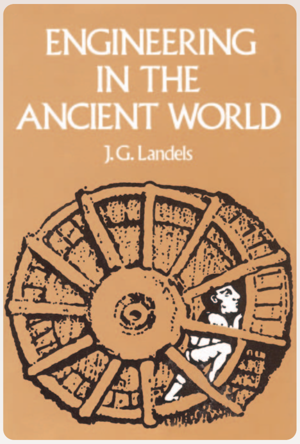
Digital Object Identifier 10.1109/MCS.2008.925740

But testing is often incomplete when the tester is also the designer or builder. Those who construct a device have already guarded against the pitfalls that they know about. It's the problems that they haven't conceived of that pose the real threat. In addition, to test something is to break it, and it's more fun to break the other guy's design than our own. Of course, the end user makes the best testing laboratory, since they're the ones who matter anyway. But asking customers to take the place of quality control might not be the best approach to consumer satisfaction. I'd like to have a warranty that allows me to charge the vendor for "testing" items I've purchased that prove to be faulty.

Figuring out what's wrong with a system isn't exactly a scientific activity, but maybe it should be. Control system testing in particular isn't yet a discipline. But some day it might be.

Dennis S. Bernstein

Inspiring Inventions



Invention I, 17 is a device for making a toy trumpet play a single prolonged note when a temple door (full-scale this time) is opened. How profound a reverence this would inspire in a worshipper is doubtful, especially if he spotted the tell-tale piece of string running from the top of the door over a pulley. II, 9 describes a thyrsus (a wand with imitation leaves at one end, emblem of the Greek god Dionysus) which whistles when thrust into water. Again, we can hardly assess the emotional effect with confidence, especially on non-Greeks. I, 21 is a coin-in-the-slot machine to be placed at the entrance to a temple which, in return for a five-drachma piece, dispensed a small amount of water for ritual washing of the face and hands. No doubt worshippers would entirely accept the need to purify themselves before entering a holy place, but their thoughts on being confronted by a sort of one-armed bandit, which stung them several days' wages for the privilege, might not have been altogether devout. Finally, Hero explains in two places (for the benefit of Romans unacquainted with Egyptian customs) that some Egyptian temples had bronze wheels mounted beside the entrance, which the worshippers turned round as they passed in, "in the belief that the bronze purifies them." He suggests two improvements on this device. One (I, 32) is a valve which causes water to spout from the hub of the wheel when it is turned, thus streamlining the two purification rituals (by bronze and by water) into one. The other, his ultimate essay in fiendish ingenuity (II, 32), was to mount the bronze wheel on the side of a box, on top of which was a little stuffed bird which, when the wheel was turned, spun round on a vertical axle and warbled. So much for the "temple miracles," and for science in the service of religion and oppression.

—J.G. Landels, *Engineering in the Ancient World*,
Barnes and Noble Books, 1978, p. 203.