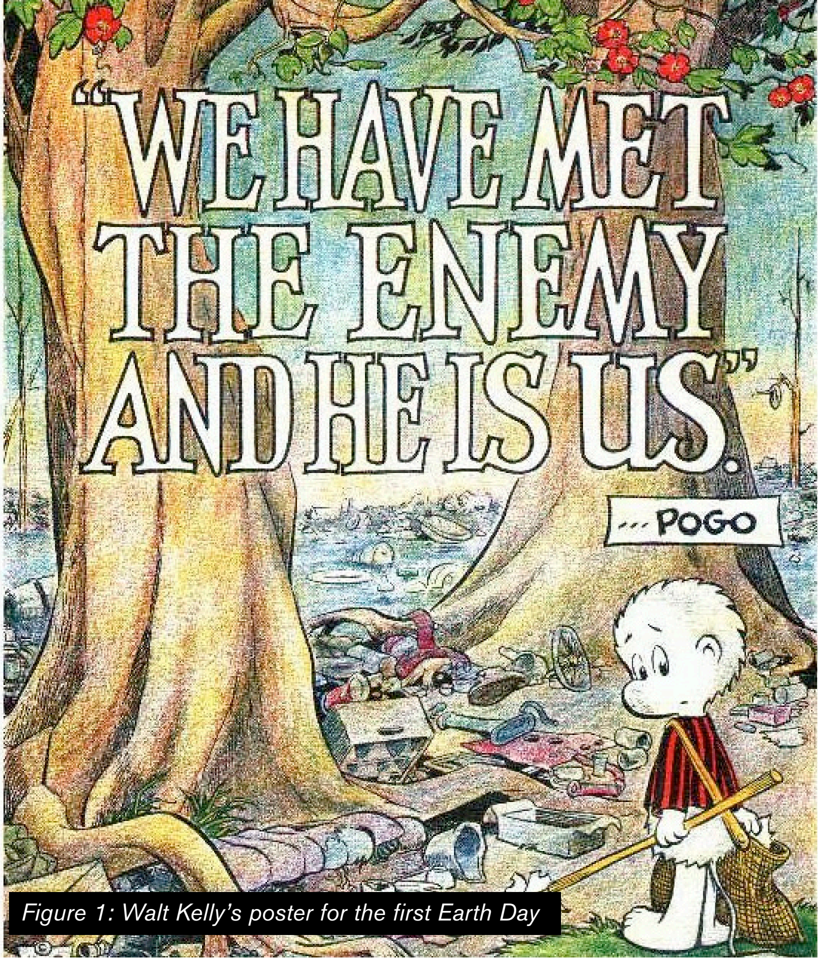

Figure 1: Walt Kelly's poster for the first Earth Day

Who Is Preying Upon Whom?

I am a frequent visitor to a local coffee vendor. While I prefer a relatively simple brew (please, give me a large cup of a caffeinated black liquid), my wife is a bit more elegant (she gets a triple venti, six pump, one Splenda, skinny hazelnut latte; or what I refer to as a diet candy bar in a cup). To save money, I have invested in a coffee card that gives me a 10% discount.

However, the other day I noticed an anomaly in the checkout system. If I use the card, I get a 10% discount. If I add money to the card, and then, on the same transaction buy coffee, it does not apply the discount.

I mentioned this to the folks behind the counter and they readily agreed that this seems to be a "known" problem. On days where an experienced counter person was working, they knew to do it as a two-step process—add money, then charge me for the coffee. However, new folks do not understand the system yet and do it all as a single transaction. The result, no discount.

In fact, the barista working last night pointed out to me that their new system had lots of "known errors." In fact, in the training class for the new system, a full afternoon was reserved for "work-arounds." The barista also pointed out that the old system, which used a non-graphical interface, worked almost perfectly. And, like generations of computer users before, upon getting the new system, they realized that the old system was much better.

Fred Brooks, in his classic (and "must read") book, The Mythical Man Month, coined the term "Second-System Effect." The second-system effect refers to the tendency of small, elegant, and successful systems to have elephantine, feature-laden monstrosities as their successors. Brooks initially used it to describe the jump from the elegantly simple operating system on the IBM 700/7000 series to the extremely complex and incredibly difficult-to-master IBM OS/360. The second-system effect occurs because designers and developers try to put in the second system all the things they did not get to do first time, loading the second system up with all the features they put off while making version one. In reality, most of these "new features" should be put off in version two (and three, four, etc.) as well. I propose a generalization of this law (and let us modestly call it "Cook's Law")—the first version of any software system can be bad. However, only subsequent versions can be truly horrible. Perhaps even this could be generalized to, "The greater the version number, the greater the chance the software is bloated and has unneeded and unusable features."

In a recent workshop at the Systems and Software Technology Conference, I got to speak on the skills necessary for a systems engineer. I used the term "The Zen of Systems Engineering." This term refers to the extreme difficulty of building a simple, elegant system. Perhaps a corollary to Brooks' Law should be that second systems tend to be bloated and feature-laden monstrosities, even when the developers are fully aware of the second-system effect. (Which, in turn, is just a restatement of Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law.)

So, what can we do to make things better? Well, for one thing, we have to aggressively work with the end users to understand what features are needed and what features are merely desired. Any time a requirement includes the words, "It would be nice if…" quit writing. Wait for the words "We have to …" Second systems will always be more complex (I have never seen a second system whose purpose was to remove features). But they should not be bloated and elephantine. Leonardo da Vinci said, "Simplicity is the ultimate sophistication." Try to put in only what the users actually need. Keep delaying the "bells and whistles" as long as you can. Make the bloated and elephantine systems somebody else's problem.

From the 1940s up until his death in the 1970s, the cartoonist Walt Kelly wrote a wonderful cartoon strip entitled Pogo. While funny in its own right, it frequently engaged in social and political satire, and therefore unless you understand the issues of the time, much of the humor is lost. In 1953, in a series of political satires attacking McCarthyism, he came up with the quote, "We have met the enemy, and he is us". This quote was also used in a poster Walt Kelly created for the very first Earth Day in 1970.

"We have met the enemy and he is us".

Wow. Kind of makes "Cook's Law" pale by comparison, huh?

P.S. I was serious. If you develop software for a living, and you have not read The Mythical Man Month, go find a copy and read it now. The Mythical Man-Month: Essays on Software Engineering is widely regarded as a classic on the human elements of software engineering.

David A. Cook, Ph.D.
Stephen F. Austin State University
cookda@sfasu.edu