Alex Groce (agroce@gmail.com), Oregon State University

Sherry Turkle's *The Second Self: Computers and the Human Spirit* concerns "the psychology of computation." Not, that is, the psychology of the computer itself (which is, thus far, a topic of science fiction, not nonfiction), but the psychology of human beings who encounter computation, who program computers, and who hypothesize about the psychology of the computer.

Some people dislike Turkle's book, for a variety of reasons. She offers interviews, generalizations, and hypotheses, in place of the more statistical and experimental mode of psychology now in vogue (there is not a p-value to be found in this book). On the other hand, those looking for philosophical abstraction, and a discussion of what is *true* about computers (or human beings, or life) will also be disappointed. Turkle uses Freudian terminology and ideas constantly, for instance. Does she believe the Freudian model (if there is such a thing as "the" Freudian model)? Well, at first it appears that she may, but eventually it becomes clear that Freudianism interests Turkle primarily as a source of metaphors. Is Freudianism correct? For Turkle, the question is not irrelevant, but less important than other questions: does Freudianism change the way we think and talk about the mind? Is Freudianism, in popular form, widely used, even when not believed? Turkle uses Freudian concepts partly because they reflect her training, but mostly because they serve as a preparation for thinking about how computational models of the mind are used by people. For Turkle, it is not critical to decide if Freudian or computational models are correct: what matters is that these models matter to people who think with them.

Why should a software engineer care? If software engineering is a purely technical activity, where complete formal specifications produced beyond our ken are delivered to us, and we construct software artifacts based on these specifications, a kind of Searle's Chinese Room development model, we shouldn't. At least, we should care no more than any other reader interested in a well-written and enjoyable book that has been much read and discussed since its original publication.

However, if we develop software in concert with users, and are ourselves users, we are forced (or invited) to consider the human element in software. How do people think about computers? How do computers change people?

The Second Self is not an HCI manual, and it will not directly help develop a better user interface. Instead, it serves as a kind of "liberal education" precursor to thinking about user issues, by simply showing the ways human beings encounter the computer, and what that encounter means.

Turkle's examples, mostly presented in interviews, quotations, and narratives, are organized into three sections. In the first section she primarily considers the first children "Growing Up with Computers," the most famous part of the book, including the stories of children using Logo. The most central question here is "is the computer alive?" A secondary concern is the difference in cultures of programming. The “hard” and “soft” mastery styles Turkle describes will be familiar to anyone who was thought deeply about the variance in ways of approaching software creation. “Hard mastery is the imposition of will over the machine through the implementation of a plan. . . soft mastery is the mastery of the artist: try this, wait for a response, try something else, let the overall shape emerge from an interaction with the medium.” This is not quite top-down vs. bottom-up, or waterfall vs. agile, but it is another way of looking at the same questions. What is the difference between the ideal Pascal program and the ideal Lisp program? Not all the stories in this section are about children; the second chapter is a look at the holding power of video games, considered in some cases in a surprisingly favorable light (a lawyer compares his gaming to Zen meditation).

The second part of the book, “The New Computer Cultures,” opens with an examination, across a span of time, of the home computer. From the first homebrew users (who, Turkle notes, are often also programmers in their day jobs) to the growth of non-technical computer users, this section looks at both individual relationships to computing and the culture of the home computer.

The examination of (mostly MIT) hackers in the next chapter of Part II dives much deeper into the idea of an actual culture built around computation. This is a culture that is much more aware it is a culture: “The hacker culture has a code. These games are for testing the mettle of the novice or for playing among master hackers.” There are rites and initiations for those who wish to join the society of those consumed by “Loving the Machine for Itself.”

Finally, Turkle’s last culture of computation is that of the philosophers of Artificial Intelligence. Here, rather than children, graduate students, or white collar workers with a hobby, Turkle considers a culture dominated by famous (at least to computer scientists) people with global ambitions on par with those of the founders of the Royal Society in the 17th century. This is an academic culture, but a less modest one than most disciplines.

These cultures are very different. The idea of “Thinking of Yourself as a Machine”, as Turkle titles the first chapter of her summarizing examination of computation, “Into a New Age,” is common: children too small to have a clear concept of whether insects are alive think of themselves as machines, at times; tenured professors with Turing Awards think of themselves as machines, quite often. Rather than seeing this as a pure threat to humane values, or a pure advancement towards a more accurate concept of the self, Turkle is most interested in simply presenting this reality.

In a critical review in the New York Times, Howard Gardner (of “multiple intelligences” fame) reviewed *The Second Self* just after its original publication in 1984. Gardner admitted that Turkle’s survey was “rich” but was disappointed Turkle never quite commits to a position on all of that richness. For Gardner, Turkle’s portrait of MIT hackers is “devastating” but she will not commit to condemning computer obsession outright, or reject the idea of computers ever experiencing emotions. Gardner wants more than a favorable nod to “the Human Spirit in a

Computer Culture” (Turkle’s last chapter); he wants Turkle to take a stand against the more extreme proponents of a computational model of self.

In the long run, Turkle’s ambivalent view is a more nuanced and useful picture than a more conclusive text, adopting either a Gardner or Marvin Minsky point-of-view. Software engineers need to operate in the knowledge that computing changes how we think, but not in any single, simple, praiseworthy or blameworthy way. The effects of encountering computation, or of creating it (and a great many of Turkle’s stories are about people who spend much of their life programming) can be simply appreciated, and seen, rather than measured, praised or condemned. Understanding is as important as appraisal.