

In the Beginning . . .

It might be said that software was open source from its earliest days. Myth-maker and journalist Steven Levy tells the story in *Hackers: Heroes of the Computer Revolution* for those who weren't there: From MIT's famously un-passworded computers to members of Silicon Valley's Homebrew Computer Club, enthusiasts were trading and actively improving on each other's software.

The first computers came with little software, and there existed no real commercial marketplace for software. People wrote their own because they needed to, and they shared their software because doing so saved everybody "the dread, time-wasting ritual of re-inventing the wheel."¹ And that sharing produced better software.²

Unlike today's networked communities, geeks of that day often worked in close proximity and could identify members of their group personally or by voice, as well as by their initials or the peculiarities of their code.

And so, without much formality, a programmer knew that sharing code was inviting others to take and reuse pieces of it or add unsolicited improvements or features.³ The best of these improvements, unsolicited or otherwise, were often called "hacks."

To qualify as a "hack," the feat must be imbued with innovation, style, and technical virtuosity.⁴

Those hacks might improve performance or add a feature, but an individual hack, in computing terms, was rarely cut from whole cloth. Programmers looked to each other's software for clues, but also for opportunities to show one's mettle. Competition among programmers was often fierce, turning into

a macho contest to prove oneself so much in command of the system that one could recog-

nize elegant shortcuts to shave an instruction or two, or, better yet, rethink the problem and devise a new algorithm which would save a whole block of instructions.⁵

Levy's history glosses what must have been some tumultuous arguments, though a hint of them can be found in Gerald Weinberg's concept of "egoless programming," introduced in his 1971 book, *The Psychology of Computer Programming*. Peer review is harsh, especially when it's misunderstood as personal criticism. Weinberger was formalizing what many programming teams had seen from the start: errors are human nature, and teamwork leads to better results faster.⁶ And teams that are, as Builder.com describes it in its Ten Commandments of Egoless Programming, "kind to the coder, not to the code," do best.⁷

And the motivation to do well, to produce great code, was in creating tools that all members of the team would use, for as Levy notes:

When you wrote a fine program you were building a community, not churning out a product.⁸

Notes

1. Steven Levy, *Hackers: Heroes of the Computer Revolution* (New York: Penguin, 2001), 41.
2. Richard Stallman, "The GNU Project" (originally published in the book *Open Sources*), GNU Operating System Web site, www.gnu.org/gnu/thegnuproject.html (accessed Mar. 19, 2007).
3. Levy, *Hackers*, 56.
4. *Ibid.*, 23.
5. *Ibid.*, 44.

6. Gerald M. Weinberg, "Egoless Programming," *IEEE Software* 16, no. 1 (Jan./Feb. 1999): 118-20.
7. Quoted in Lamont Adams, "Download the Builder.com Ten Commandments of Egoless Programming," June 6, 2002, Builder.com Web site, <http://builder.com.com/5100-6404-1045782.html> (accessed Mar. 19, 2007).
8. Levy, *Hackers*, 56.