

Linux is subversive. Who would have thought even five years ago (1991) that a world-class operating system could coalesce as if by magic out of part-time hacking by several thousand developers scattered all over the planet, connected only by the tenuous strands of the Internet?

Certainly not I. By the time Linux swam onto my radar screen in early 1993, I had already been involved in Unix and open-source development for 10 years. I was one of the first GNU contributors in the mid-1980s. I had released a good deal of open-source software onto the Net, developing or co-developing several programs (nethack, Emacs's VC and GUD modes, xlife, and others) that are still in wide use today. I thought I knew how it was done.

Linux overturned much of what I thought I knew. I had been preaching the **Unix** gospel of **small tools, rapid prototyping, and evolutionary programming** for years. But I also believed there was a certain critical complexity above which a more centralized, *a priori* approach was required. I believed that the most important software (operating systems and really large tools like the Emacs programming editor) needed to be built like cathedrals, carefully crafted by individual wizards or small bands of mages working in splendid isolation, with no beta to be released before its time.

Linus Torvalds's style of development—**release early and often, delegate everything you can, be open to the point of promiscuity**—came as a surprise. No quiet, reverent cathedral-building here—rather, the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches (aptly symbolized by the Linux archive sites, which would take submissions from *anyone*) out of which a coherent and stable system could seemingly emerge only by a succession of miracles.

The fact that this bazaar style seemed to work, and work well, came as a distinct shock. As I learned my way around, I worked hard not just at individual projects, but also at trying to understand why the Linux world not only didn't fly apart in confusion but seemed to go from strength to strength at a speed barely imaginable to cathedral-builders.

Freedom is not an abstract concept in business.

The success of any industry is almost directly related to the degree of freedom the suppliers and the customers of that industry enjoy. Just compare the innovation in the U.S. telephone business since AT&T lost its monopoly control over American consumers with the previously slow pace of innovation when those customers had no freedom to choose.

The world's best example of the benefits of freedom in business is a comparison of the computer hardware business and the computer software business. In computer hardware, where freedom reigns for both suppliers and consumers alike on a global scale, the industry generates the fastest innovation in product and customer value the world has ever seen. In the computer software industry, on the other hand, change is measured in decades. The office suite, the 1980s killer application, wasn't challenged until the 1990s with the introduction of the web browser and server.

Open-source software brings to the computer software industry even greater freedom than the hardware manufacturers and consumers have enjoyed.

Computer languages are called languages because they are just that. They enable the educated members of our society (in this case, programmers) to build and communicate ideas that benefit the other members of our society, including other programmers. Legally restricting access to knowledge of the infrastructure that our society increasingly relies on (via the proprietary binary-only software licenses our industry historically has used) results in less freedom and slower innovation.

Open source represents some revolutionary concepts being thrown at an industry that thought it had all of its fundamental structures worked out. It gives customers control over the technologies they use, instead of enabling the vendors to control their customers through restricting access to the code behind the technologies. Supplying open-source tools to the market will require new business models. But by delivering unique benefits to the market, those companies that develop the business models will be very successful competing with companies that attempt to retain control over their customers.

There have always been two things that would be required if open-source software was to materially change the world: one was for open-source software to become widely used and the other was that the benefits this software development model supplied to its users had to be communicated and understood.

I just referred to the “open-source movement”. That hints at other and perhaps more ultimately interesting reasons for the reader to care. The idea of open source has been pursued, realized, and cherished over those thirty years by a vigorous tribe of partisans native to the Internet. These are the people who proudly call themselves “hackers”—not as the term is now abused by journalists to mean a computer criminal, but in its true and original sense of an enthusiast, an artist, a tinkerer, a problem solver, an expert.

The tribe of hackers, after decades spent in obscurity struggling against hard technical problems and the far greater weight of mainstream indifference and dismissal, has recently begun to come into its own. They built the Internet; they built Unix; they built the World Wide Web; they’re building Linux and open-source software today; and, following the great Internet explosion of the mid-1990s, the rest of the world is finally figuring out that it should have been paying more attention to them all along.

The hacker culture and its successes pose by example some fundamental questions about human motivation, the organization of work, the future of professionalism, and the shape of the firm—and about how all of these things will change and evolve in the information-rich post-scarcity economies of the 21st century and beyond. The hacker culture also, arguably, prefigures some profound changes in the way humans will relate to and reshape their economic surroundings. This should make what we know about the hacker culture of interest to anyone else who will have to live and work in the future.