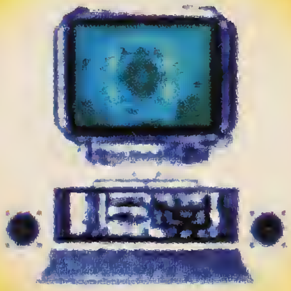
THE UPDATED CLASSIC

fire in the valley



THE MAKING OF THE PERSONAL COMPUTER

SECOND EDITION

PAUL FREIBERGER & MICHAEL SWAINE

fixed in this open atmosphere and how rapidly technology can advance when anyone is free to contribute to the most promising projects, what are they talking about but bootstrapping?

The goal that Engelbart set for himself nearly 50 years ago still eludes him. There have been many quiet successes, a few moments of acknowledgment, off-again-on-again funding, and much disappointment. But with the embracing of the Internet by businesses, we may be moving toward the future that Engelbart envisioned so long ago. He is guardedly hopeful, but he continues to work 12 hours a day at his Bootstrap Institute, seeking ways technology can make people smarter.

Citizen Nelson

I thought there would be a real computer revolution; I see complete betrayal.

Ted Nelson
The Thomas Paine of the personal computer revolution

When the computer industry came together in 1998 to honor Doug Engelbart at the "Engelbart's Unfinished Revolution" event, it was only fitting that Ted Nelson be there. Not because he had built any machines or based any software on Engelbart's ideas. He had not. Nelson was, in fact, a seer in his own right, the begetter of a vision every bit as grand and detailed as Engelbart's. With Engelbart and Nelson on stage, the audience was viewing the prophets of two rival religions.

Although he praised Engelbart lavishly, Nelson also identified the key difference in their philosophies. "I visited Doug in the spring of 1967," he told the audience. To him, Engelbart's emphasis on collaboration "seemed completely naive. I've always been sensitive to conflict: agreement among people is a miracle. One of the things that moves me greatly is seeing this audience here because it shows that [Engelbart's] emphasis on collaboration, on working together, does have meaning."

Nelson, though, has always been more interested in how to empower the dissenter. This is not surprising, since he is often described as the Thomas Paine of the personal computer revolution.

The son of Academy Award–winning actress Celeste Holm and director Ralph Nelson, Ted Nelson felt the lure of show business early on, yet was always deeply impressed by the potential of computers. Like Ada Byron, Lady Lovelace, he sought a synthesis of art and technology.

In 1974, before the Altair was announced, before there was anything that could be accurately called a personal computer, he self-published *Computer Lib,* in which he explained computers to the lay reader in clear and witty prose and laid out the political agenda for the personal computer revolution.

"You can and must understand computers NOW," the book trumpeted. "Computer power to the people! Down with cybercrud!" Also, "If computers are the wave of the future, displays are the surfboards." Along with the bumper-sticker slogans were many essays, jokes, thoughts, and anecdotes, all stitched together in a style blatantly imitative of Stewart Brand's underground classic *The Whole Earth Catalog.*

The book somehow reached all the right people. It influenced Steve Wozniak. Ed Roberts had it on his desk while MITS was building the Altair. *Computer Lib* was Thomas Paine's *Common Sense* for the computer hobbyist. Lee Felsenstein wrote, "Ted succeeded with *Computer Lib* to rally a rabble of latent crackpots into an anarchistic army which breached the sanctum of Official Computerdom and brought computers to everyone."

The book was fiercely individualistic in style and in philosophy. Because there were no personal computers, Nelson urged people to turn what technology was available to their personal use. "Maybe you should consider buying your own minicomputer [or] chipping in with several other families to get one." He gave contrarian business advice that served the early personal computer makers well: "This is . . . a field where individuals can have a profound influence. But the wrong way to try it is through conventional corporate financing. [D]o it in a garret, and *then* talk about ways of getting it out to the world."

Nelson spoke as provocatively and as entertainingly as he wrote, and was much in demand as a speaker at computer conferences. In April 1977, he addressed the West Coast Computer Faire on the subject of "Those Unforgettable Next Two Years." In the speech, he correctly predicted that major technology companies would enter the field, but would have trouble because of their cumbersome decision-making processes; that standards like the S100 bus would dictate which computers succeeded and which didn't; that small programs would give way to large, complex programs requiring more memory; and that the legendary and unassailable IBM would run into big problems, leading to massive layoffs and reorganization.

He was just continuing a record of prescience: in *Computer Lib* he had predicted the arrival and the amazing spread of personal computers, and the transfer of most office work from paper on desks to computer desktops.

With the arrival of the revolution he had predicted and fomented, however, Nelson had little patience with the pace of progress or the actual systems.

He rebelled at things as fundamental as the structure of files on a computer, even the concept of files. To him, file structures impose a hierarchical mindset that has nothing to do with reality. You put away the current item neatly and pry open the packing crate in which the next is stored. "Today's software is designed for clerks and engineers," he said, "not for people who think."

Software applications also ticked him off. "Word processing" is not a category of human activity, he fumed, but a label for the artificial boundaries imposed by inadequate programs. He found most application programs disappointing, but he was more offended by the very concept of application categories such as word processing. The idea that one should be boxed into these tight categories of computer use angered him.

The acronym WYSIWYG, for What You See Is What You Get, was supposed to convey the boons of new printing technology in the 1980s. "What it really means," Nelson complained, "is What You Get When You Print It Out. In other words, we are using the computer as a paper simulator, which is like tearing the wings off a 747 and driving it as a bus on the highway."

Nor did he believe the Macintosh was the panacea some claimed. He called it "an Application Prison, distracting its inmates with bread and circuses (Fonts and Graphics) from the fact that there was no structural support for organization of projects. And it gave us the Abominable Hidey-Hole—called 'The Clipboard'—except that you can't see it, it holds only one item, and each item destroys the previous; in ALL other respects like a regular clipboard, except there aren't any [other aspects]. This is called a 'metaphor,' meaning a stupid scrap of resemblance on which bad software is built."

The whole software development process was wrong, he was convinced. His take on the writing of Lotus Symphony: "Mitch said, 'Here's the Lotus wish list, will you implement exactly as stated?' [The programmer] programmed it and got backing to develop Lotus Notes and the list was released as Lotus Symphony." The tale may be apocryphal, but it did have the ring of truth.

Software, Nelson was sure, should be "designed according to an auteurist approach—an approach already working quite well in video games—integrated virtuality unified by the hand of a movie director, who trades off, polishes, and edits all the effects. The incredible clunkiness of today's office software begs for huge speedup and a new silky feel.

"Why are video games so much better designed than office software? Video games are designed by people who love to play video games. Office software is designed by people who want to do something else on the weekend."

He's serious about the movie director analogy. "Interactive software is a branch of moviemaking. And most computer science is irrelevant. What is relevant is studying your Orson Welles, studying your Alfred Hitchcock, [and] studying your good documentaries. Because right now we are in the stage of software which compares to the movie business before 1904, where movies were made by the cameraman because he understood the equipment. In 1904, they invented the director. He had to know how to make those parts come together."

As the son of a movie director, Nelson tried his own hand at software design. As a college student at Harvard, Nelson generated huge amounts of notes and began to despair of ever getting them all organized. Then in 1960, he discovered computers and had an immediate epiphany. I'll just write a program to keep track of all my notes, he thought. Nearly 40 years later, he's still working on it.

Certain features seemed necessary. The program should reflect the way thoughts are organized. For one thing, thinking is parallel, not linear. Furthermore, he didn't want to throw away old versions of a document when he produced a new one; he wanted the various versions linked together. Footnotes are a feeble way to tie ideas together, no more helpful than a fingerpost. Surely in moving from paper to electronic storage, one could find a better approach. But links should go in both directions, so that he could see what documents referred to this document, as well as what documents it referred to. He called the system "hypertext."

With the help of many programmers over several decades, these ideas evolved into a massive project called Xanadu.

Xanadu would be a world of interconnected text, graphics, sound, and video, like the World Wide Web, but unlike the Web it would provide version management, links that worked both ways and didn't break, transparent compensation for authors, and support for expression that recognized the nonlinearity of thought. In short, the World Wide Web done right (and conceived long before).

For years, Nelson and his programmers labored to bring Xanadu to life. They worked generally without adequate funding, pursuing a goal that, like Engelbart's plans, kept moving further off. For a while, Autodesk, a successful public software firm in San Rafael, north of the Golden Gate Bridge, backed the project, and it looked like it was going to happen. Then a few years later, the Web sprang to life and popularity. Like the Samuel Taylor Coleridge poem for which it was named, Xanadu remains unfinished.

"Getting [financial] backing," he explains, "especially to do things *way* differently, is exactly like trying to get backing for a movie. It's complicated politics, everybody thinks he can direct, and that's what Hollywood sorts out—the politics of Getting to Direct." He hasn't gotten to direct yet; like Engelbart, he hasn't realized his grand design.

In September 1999, Nelson either admitted defeat or took a logical and fitting step in the evolution of Xanadu: he released its unfinished code under an open-source license. In the future the implementation of the Xanadu dream would be something to which anyone, rather than just the small team of programmers for whom Nelson was able to find funding, could contribute. For better or worse, its future was now out of Ted Nelson's hands.

Speaking of Engelbart, he said, "We both made the same fundamental error, I think—what each of us was trying to do required the existence of the system we were building in order to build it. He seems to want his 'high-power teams' on hand to recognize and choose the structures to be created in order to create and empower high-power teams. Similarly, I have assumed the existence of [my kind of hypertext] literary systems in order to manage the versions of trying to build them."

The computer systems that exist today, as well as the World Wide Web, represent a failure to Nelson, who holds firm to his more ambitious vision.

"I see a completely different computer world, of empowerment and putting real computer power in people's hands. Forms of publication that are orders-of-magnitude richer in structure and comparison, rather than dumb special effects. So what I still intend to do: the same, retrofitted to the Web. Another day will dawn, I tell you. And it will NOT be stupid Talking Agents on a screen." Like Douglas Engelbart, Ted Nelson is still working to bring about the *real* computer revolution that he sees so clearly and that *still* hasn't happened. Like Mitch Kapor, he has always been keenly aware that technological decisions have political consequences. Like Steve Wozniak, he believes that the goal must be the empowerment of the individual human being.

From Nelson's impassioned "Computer power to the people!" in *Computer Lib* and David Ahl's earnest pitch to DEC management to build a personal computer, through the wild enthusiasm that greeted the announcement of the Altair, to the ubiquity of personal computers in homes, schools, and offices today, and the ongoing transformation of business and society by the Web, empowering the individual has been a powerful motivator. Indeed, for many who wrought the revolution, that was what it was all about. Taking that as the goal, it is clear that we have come a long way. And we have a long way to go.

The revolution is over, and it has only begun.

BUSINESS/TECHNOLOGY

> "A great adventure that gives the reader a sense of being close to a historical movement that is still playing itself out."
>
> —*From the foreword by John Markoff, The New York Times*

In January 1975, *Popular Electronics* magazine published a cover story on the Altair, an odd metal box with switches and blinking lights that proved to be the progenitor of today's personal computer. Inspired by possibilities that the leaders of the electronics and mainframe computer industries couldn't see, unlikely entrepreneurs—hippies, dropouts, phone phreaks, and electronics hobbyists—seized the opportunity.

How those personal computer pioneers went from side street garages to Wall Street's graces, and how their brilliance, enthusiasm, camaraderie, and competition changed the world is all here in Paul Freiberger and Michael Swaine's classic, *Fire in the Valley: The Making of the Personal Computer.* First released in 1984, it uniquely captures the explosive, frenetic energy of those early days.

This updated edition features interviews with the major players, new chapters, dozens of new photos, and updates throughout that carry the story forward into the Internet era. The authors convey the exciting development of companies such as Apple, Microsoft, Sun, Netscape, Lotus, and Oracle.

Itself a milestone in the fascinating history of the personal computer, *Fire in the Valley* is the definitive account of how it all happened and why.

Paul Freiberger is the coauthor of *Fuzzy Logic*, winner of the 1993 *Los Angeles Times* Book Prize, and has written for the *San Jose Mercury News*, the *San Francisco Examiner*, and National Public Radio. He currently works at the Interval Research Corporation in Palo Alto.

Michael Swaine is editor-at-large for *Dr. Dobb's Journal*. He is also a popular columnist for print and electronic magazines in the United States, Italy, and Germany, and maintains *Swaine's World*, a Web site that tracks computer industry news, at www.swaine.com.

McGraw-Hill
A Division of The McGraw-Hill Companies

$16.95 U.S.A.

ISBN 0-07-135892-7

90000

9 780071 358927

www.fireinthevalley.com
www.books.mcgraw-hill.com

6 39785 31773 9

Cover Design: 12E Design