

Historical Reflections

The Immortal Soul of an Old Machine

Taking apart a book to figure out how it works.

THE BEST BOOK ever written about IT work or the computer industry will be 40 years old in August. Tracy Kidder's *The Soul of a New Machine* describes the work of Data General engineers to prototype a minicomputer, codenamed "Eagle," intended to halt the advance of the Digital Equipment Corporation's hugely successful VAX range. It won both the Pulitzer Prize and National Book Award for non-fiction, perhaps the two highest honors available for book-length journalism. Year after year, the book continues to sell and win new fans. Developers born since it was published often credit it with shaping their career choices or helping them appreciate the universal aspects of their own experiences.

Soul's appeal has endured, even though what started out as a dispatch from a fast-growing firm building a piece of the future now reads as a time capsule from a lost world. Back in 1991 I read the book for an undergraduate class, typing my paper on a PC that was already more capable than Eagle yet cost 100 times less. So why are so many people still excited to relive the creation of a pitifully obsolete computer, designed by a team of obscure engineers for a long-forgotten company that never mattered very much anyway? Having spent almost 30 years now trying to take the book apart and figure out how it works, I think I have some answers. Ten of them, in fact.



1: It Does Not Assume You Know Anything

Paradoxically, the obscurity of Data General helps to explain the book's enduring power. My shelves are full of books about Microsoft, Apple, Netscape, and Oracle written while the companies were famous. Their authors assumed anyone who picked the book up was already fascinated with the company, cared deeply about its products, and would enjoy endless pages of gossip, corporate strategy, legal maneuverings, and trivia. They have not aged well.

In contrast, *Soul* delivers a self-contained package, containing everything you need to enjoy the story. Back in 1981 most potential readers had never used a computer of any kind, still less a "super minicomputer." Kidder would have had to self-publish any book written for Data General fans, and anyway he knew nothing about computers when he arrived in Westborough, MA, to follow up a suggestion from his editor. Kidder's only previous book was about a murder and his main life experience, other than a Harvard degree in English and

an MFA from Iowa, was having spent two unhappy but uneventful years as First Lieutenant, Military Intelligence in Vietnam.

2: It Reads Like a Classic American Novel

The book was a milestone in the development of what is now called “literary nonfiction.” To keep us turning pages, Kidder draws deeply on the mythic archetypes of American literature, as in his introduction of Tom West, his protagonist and the project’s leader, during a prologue that recalls Ernest Hemingway: West awes the other crew members of a small sailing boat with his stamina and ruggedly taciturn optimism when hit by a storm. “Whatever he did for a living,” they conclude, “it was probably interesting and obviously important.”

Having hooked us on the enigmatic Tom West, Kidder is cocky enough to spend an entire chapter without mentioning him, instead introducing Data General as “the Darth Vader of the computer industry” (a reference that undoubtedly aged better than he expected). Data General’s corporate culture was defined in equal parts by thrift and aggression. Kidder confides that Data General’s spartan corporate offices were engineered for rapid conversion to factory space even before he gets around to mentioning the lawsuit a rival firm filed to accuse Data General of burning down its factory.

When West reappears, Kidder stimulates our curiosity by presenting him as a figure of mystery to his own team members: a CIA agent, a folksinger, a speed freak, even “a prince of darkness.” I was struck by the similarities with the technique F. Scott Fitzgerald used in *The Great Gatsby* to introduce his title character. Jay Gatsby, urbane host of fabulously swanky parties, turned out (spoiler alert) to be plain old James Gatz of North Dakota, a lovelorn bootlegger desperate to recapture the attention of his lost sweetheart. Less dramatically, corporate computer engineer Tom West, turns out to be Joseph Thomas West III, an engineer from a privileged background who came late to corporate life after taking a year off during college to play folk music, followed by seven slightly offbeat years building and delivering digital clocks for the

Many readers were tantalized by the idea of computer architecture as a creative medium in which experts could read traces of individual flare or, as here, a conservative organizational culture.

Smithsonian Observatory. That doesn’t quite justify the build-up, but whatever: we are already hooked. Confessing to his editor that he was having difficulty capturing West, “whose special vanity had been to make himself mysterious to me as well as to his team of computer engineers,” Kidder had been advised to “do a Gatsby on him.”⁴

3: It Roots for the Underdogs

Gripping stories often ask us to root for underdogs to triumph against the odds. Data General’s brutal corporate culture gave West space to launch his project but deprived his team of resources, leaving them in cramped and uncomfortable conditions. Even pencils were in short supply. If you are thinking that a major minicomputer firm ought to have been able to provide pencils to the elite team building its next-generation system, you’d be right. However, that team and its pencils were down in North Carolina, designing an ambitious all-new 32-bit architecture on a clean sheet of paper. Data General had caught a bad case of what Fred Brooks called the “second system syndrome.”⁴ The tactic of throwing out backward compatibility for a new architecture worked for IBM with its legendary System/360 gamble, but has failed far more often, for example with IBM’s own Future Systems project, at least three times for Intel (iAPX 432,

80960, and Itanium) and even DEC’s much-admired Alpha processor.

The clean sheet approach failed for Data General too. Anticipating this, Tom West rounded up the best of the engineers left behind in Massachusetts to launch a semi-clandestine effort to produce a 32-bit extension of the existing 16-bit Eclipse minicomputer, preserving compatibility by interleaving old and new instructions seamlessly rather than using a “mode bit” to enter a separate legacy mode. (More than 20 years later, AMD played a comparable trick on Intel by extending the standard x86 architecture to 64-bits). For the project to be approved as “insurance” against problems with the North Carolina team, West had to promise the impossible task of producing the entire computer in one year. Thirty engineers crowded together, turning the basement of Data General’s headquarters into a site of relentlessly hard work.

The team’s outsider triumph gives way to an unexpectedly mournful conclusion. West himself is banished to a marketing job in Japan. Kidder, winking at his name, compares West to a gunslinger, who dispatches the bad guys only to be run out of town by the very citizens he saved. “It was a summer romance,” realizes West. “None of it came out the way he had imagined it would, but it was over and he was glad.”

4: It Captures the “Crunch” of Startup Development

Although Data General was a mature company, the project was run more like a startup. West and his lieutenants staffed the team with young men fresh from engineering school, lured with the prospect of being able to design a new computer architecture. They boasted of being “a place where people are really doing the next thing” but cautioned that “there’s a lot of fast people in this group ... a real hard job with a lot of long hours.” In short: “tell him that we only let in the best. Then we let him in.” It’s a classic example of what software engineering writer Ed Yourdon called the “marine corps” justification for a “death march project.”⁷ As Kidder put it, “It was kind of like recruiting for a suicide mission. You’re gonna die, but you’re gonna die in glory.”

Eagle is brought to life more slowly than the team had promised but sooner than Data General had any right to expect. Inexperienced recruits were manipulated into “signing up” to aggressive schedules, because an unreasonable commitment given freely motivates more deeply than one imposed by management. “Signing up required, of course, that you fervently desire the right to build your machine and then you do whatever was necessary for success, including putting in lots of overtime, for no extra pay.” The novice engineers are granted large responsibilities and the freedom to follow their instincts. Young men “dribble away” pieces of their lives as they battle to prove themselves. Some wilt under the pressure; those that remain work frantically and effectively. The hardware team (“The Hardy Boys”) and the microcode developers (“The Mikrokids”) battle constantly and informally against each other to add and remove hardware capabilities from the specification. Together, they take the computer from conception to prototype hardware in six months. Then they have to make it work.

What was the “soul” referenced in the book’s attention-grabbing title? Natalie Angier, an early reviewer, claimed that the “soul of a new machine, says Kidder, is nothing more than the collective soul of those who put the machine together.” That’s plausible, though Kidder notably declined to “say” this directly. The closest he comes, which is not very, is describing a home workshop, full of power tools and carpentry equipment, as “a window on West’s soul”? Nevertheless, his title made me think of the old story of the Golem, animated by magic but created by, and enslaved to, human will. As one of the engineers explained, “I don’t have to get official recognition for anything I do. Ninety-eight percent of the thrill comes from knowing that the thing you designed works, and works almost the way you expected it would. If that happens, part of *you* is in that machine.” Writing on the front page of the *New York Times Book Review*, Samuel C. Florman called the engineers “fanatics, but not purists.”³ Perhaps their many individual sacrifices, of marriages, mental stability, youth, and health, amounted to a ritual through which sundered fragments of

their own souls accumulated to bring the new machine to life.

5: It Gets to Technology Through People

How to make a reader commit to almost 300 pages of finely observed business history of a company they hadn’t heard of, focused on the creation of a computer they would never see, interspersed with technical descriptions of microcode, caching, and instruction formats that might have been more at home in a computer architecture textbook than a gripping best seller? Kidder succeeds by telling us first about people, not about machines, investing us enough in them and their work to follow as he moves deep into descriptions of the problems they were grappling with. People have changed much less than computers over the last 40 years so this material remains gripping today.

When reading or watching the stories of the computer industry’s most successful men we know all the endings. Each retelling of the story of Steve Jobs is like riding a rollercoaster: we hurtle along a fixed track past expected triumphs and tragedies. In contrast, Kidder tells an unfamiliar story, tightly focused in time and place yet larger than any of its individual players.

In the book’s most famous passage, West has a friend sneak him into a data room where a VAX is installed to get a feel for the machine he is trying to beat. Lifting the covers of its central processing unit he counts 27 printed circuit boards: “Looking into the VAX, West had imagined he saw a diagram of DEC’s corporate organization. He felt that VAX was too complicated. He did not like, for instance, the system by which various parts of the machine communicated with each other; for his taste, there was too much protocol involved. He decided that VAX embodied flaws in DEC’s corporate organization. The machine expressed that phenomenally successful company’s cautious, bureaucratic style.”

Many readers were tantalized by the idea of computer architecture as a creative medium in which experts could read traces of individual flare or, as here, a conservative organizational culture. Most chapters tell us a piece of the story from the viewpoint of engineers responsible for design-

ing a part of the machine or for running the debugging process, introducing a rich cast of clearly delineated supporting characters with their own quirks and motivations.

One such chapter, “The Case of the Missing NAND Gate,” begins by introducing several engineers. Kidder sketches the lives, habits, and appearances of Ken Holberger (“Chief Sergeant Detective of the Hardy Boys” who “couldn’t look messy if he tried” but “doesn’t waste time listening to people who aren’t making good, relevant sense”), Jim Veres (whose “stern glare ... makes some people nervous. His managers’ confidence in him is tempered only by their feeling that he works too hard. That is how they express it”) and Jim Guyer (an asthmatic mountaineer who “seems in his busyness, among the happiest of the group”). Kidder follows their interactions while troubleshooting a problem, observing the feelings of each toward possible flaws in their own boards. We care about the bug because we see how much it matters to these engaging characters and to the unseen narrator who leads us confidently through passages such as “The diagnostic program originally puts the target instruction at address 21765, and then, sometime later on, it moves the target instruction to 21766. But the IP never gets word of the change, though the System Cache does.”

By the end of the book we know about microcode, bits and bytes, Boolean algebra, what happens when an instruction is executed (in some detail), memory management, debugging, diagnostics, emulation, and the Adventure game. In his *Times* review, Florman noted that while these descriptions “did not significantly increase” his “own very superficial knowledge” the “uninitiated will find these brief passages abstruse but not bewildering, unfathomable but not boring.”

Kidder’s ability to hold our interest is aided by another structural similarity with Scott Fitzgerald’s masterpiece: a narrator defined primarily by his obsession with the man of mystery. Kidder acknowledges his own ongoing presence with phrases such as “I saw them all collected once ... during a fire drill,” or “West said years later,” or “I saw him at one of the team’s parties” but refuses to make himself a character.

That is a contrast with the work of flashier proponents of 1970s “new journalism” such as Tom Wolfe, author of vivid accounts of the American space program and the worlds of car enthusiasts, or Hunter S. Thompson who made his own erratic behavior the center of every story. Kidder has called this the “first-person minor” or “reasonable person” technique of narration, in which “not much about the narrator is revealed, including the narrator’s opinions.”⁴ Stripping his in-book presence of most identifying marks, to leave a person-shaped avatar, helps us to imagine ourselves in Kidder’s empty shoes as they move through the basement or follow West home. We come to identify with the narrator’s only defined characteristics: initial ignorance, growing fascination, and dogged pursuit of understanding.

6: It Exposes the Materiality of Computing

Kidder calls Data General’s products “machines” as often as “computers” and chose the word “machine” for his title. In 1979, computers were built on a scale where engineers could probe and rewire each logical pathway, giving Kidder something material to describe. That is a contrast to our current discourse in which “the digital” is assumed to be invisible and immaterial. Early minicomputers like DEC’s PDP-8 and Data General’s Nova, cheap and small in comparison to mainframes, were made possible by a stream of innovations to package and assemble electronics more efficiently. By 1979, the Eagle team was building computer logic mostly out of chips, rather than discrete transistors and resistors. Yet despite its innovative use of Programmable Array Logic chips for custom logic the Eagle’s central processing unit still filled many circuit boards.

Kidder emphasizes continuities between tinkering with broken machines, a common activity in the 1970s, and the work of the engineers as they closely observe Eagle’s functioning with logic probes, adding wires or tweaking circuits to fix tiny errors in the design. West boasted “I can fix anything,” which Kidder documents for a diesel engine, televisions, clocks, furniture, a record player, and a house. “What that

Further Reading

I will be tackling two other landmark studies of the culture of IT work, Steven Levy’s *Hackers* and Ellen Ullman’s *Close to the Machine* in later columns, along with the recent TV series *Halt and Catch Fire*. In the meantime, *Soul* may leave you hungry for closely observed studies of development projects.

I can recommend Fred Moody’s *I Sing the Body Electronic* (Viking Penguin, 1995), about a year spent with a Microsoft group developing multimedia CD products, and G. Pascal Zachary’s *Show Stopper!* (Free Press, 1994) about the creation of Windows NT. Michael Lewis is an outstanding writer. Sadly *The New Thing* (Norton, 1999), which follows Netscape co-founder Jim Clark as he pitches half-baked healthcare marketplace and builds a giant robot yacht, is not one of his best but it is still readable and insightful.

Writers have seemed less likely to embed themselves in computer projects recently, perhaps because companies have become increasingly secretive. *Masters of Doom* (Random House, 2003) is based on interviews rather than direct observation but still gives a vivid account of the testosterone-soaked development of breakthrough action games *Doom* and *Quake*. Jason Schrier’s *Blood Sweat and Pixels* (Harper Paperbacks, 2017) is a collection of grim short stories versus Kidder’s elegiac novel, with each chapter looking at a different video game project.

Thomas A. Bass’s *The Eudaemonic Pie* (Houghton Mifflin, 1985) tells the inside story of an eccentric Californian team building shoe-mounted computers and radio links to predict the destination of a ball bouncing on a roulette table. For memoirs focused on product-based startups, try Jerry Kaplan’s *Startup* (Houghton Mifflin, 1994) about a pen computing pioneer crushed by Microsoft, Charles Ferguson’s judgmental *High Stake, No Prisoners* (Crown Business, 1999) about a web editing package sold to Microsoft, and Michael Wolff’s *Burn Rate* (Simon & Schuster, 1998) in which a gossipy journalist describes his largely unsuccessful efforts to obtain millions of dollars from venture capitalists. Reflecting the mood of their era, all three focus more on finance and management than on engineering.

thing was,” Kidder continues, “whether a car’s engine of a computer, did not matter; but since computers were among the most complex of all man-made things, they had seemed to him, he said, to pose interesting challenges.” The main story ends when the prototype is “wheeled down the hall to Software.” Kidder barely mentions this less-tangible side of the project, which accounted for more than half of the total development work.

A single Eagle would sell for a quarter of a million dollars and could support dozens of simultaneous users, each on a separate video terminal. Mass-produced computers, including the Apple II and TRS-80, had been sold to consumers and hobbyists since 1977. But *Soul* ignores them, as did DEC and Data General during that period. Personal computers still seemed like toys, and the chip technology of the era was several years away from being able to create a high-performance 32-bit microprocessor. More fundamentally, the shift to standard processors stripped the heart out of computer engineering. A few years later, West spoke dismissively of “all these people who are putting 68000s on a board and calling it a computer.”⁵

7: It’s Unashamedly Masculine

This is, as you have surely already realized, a remarkably and unself-consciously masculine book—which helps it appeal to men and to those comfortable with the dominant traditions of American literature. While computers were new and unfamiliar, American literature had a long tradition of celebrating the rugged masculinity of civil engineers taming the Western landscape and the resourcefulness of pioneers able to fix or adapt machinery to their needs. Such men were also expected to be taciturn, emotionally restrained, and hard to know. Perhaps overcompensating for the growing association of computing with nerds, Kidder celebrated men doing guy things to an extent that must have seemed old-fashioned even in 1981.

West draws an organization chart on a whiteboard, then puts an X over a rival manager, saying “This guy disappears in time.” The Eagle group themselves come to feel like throwbacks to an earlier, less bureaucratic kind of engineering, joking about ordering dinosaur T-shirts for the team and complaining that “beating people up didn’t seem to get results anymore.” After all the mythologizing, it was a

Distinguished Speakers Program

A great speaker can make the difference between a good event and a WOW event!

Students and faculty can take advantage of ACM's Distinguished Speakers Program to invite renowned thought leaders in academia, industry and government to deliver compelling and insightful talks on the most important topics in computing and IT today. ACM covers the cost of transportation for the speaker to travel to your event.

speakers.acm.org



Association for
Computing Machinery

Computing was a sideshow 40 years ago. Today Apple, Amazon, Microsoft, and Alphabet are the first trillion-dollar companies. Technology and money are inseparable.

shock when I found a YouTube video of West helping to introduce a successor to the Eagle in 1990: an apparently charisma-free middle aged engineer droned through technical specifications and corporate jargon, sweating in his short-sleeve shirt and tie.

Their jobs required them to sacrifice or downplay any commitment to family or human relationships. Kidder mentions in passing that one female engineer was hired, Betty Shanahan. We learn only that her husband was unhappy to be left doing the laundry and that she was given a joke award for “putting up with a bunch of creepy guys.” Eventually Shanahan got tired of putting up, becoming an advocate for diversity as executive director of the Society of Women Engineers. Today there might still be only one woman on the team, but a modern author would surely center a chapter on her.

We learn far more about Rosemary Searle, the project's secretary and surrogate mother to its young men. She tells Kidder that West “never put one restriction on me ... he let me go out and see what I could get done.” When I wrote about the creation of ENIAC, Kidder's sensitivity to Searle's contributions reminded me to highlight what little information I could find on Isabelle Jay, its secretary and its longest serving full-time member.

8: It Dramatizes Ordinary Engineering Work

Eagle fared well after it reached market, as the Eclipse MV/8000. It and its successors sustained Data General for years. The company was saved, kind of, though it never returned to its glory days of industry leading growth rates and profit margins. Before long the entire minicomputer industry was crumbling. In 1999, Data General was gobbled up by EMC for its storage technology. By then even DEC, once second only to IBM, had concluded its slow decline with absorption by PC manufacturer Compaq. In time, all empires turn to dust.

The work of these engineers is challenging and difficult to understand, but Kidder treats them as skilled practitioners of a difficult craft rather than world-shaking genius innovators. You might complain that Kidder wasted his talent on the wrong story, that he should have spent the late-1970s lurking in a garage in Silicon Valley rather than a basement in Massachusetts. If you have heard of Steve Wallach, Carl Alsing, Ed Rasala, or even Tom West himself it is almost certainly because of their appearance in the book, despite their many accomplishments at firms like Convex and Alliant. Personally, I am glad Kidder told the story he did, looking at a part of the computer industry that was far larger at the time and remains more representative of engineering practice.

Novelists know that ordinary lives are full of hidden drama, but most technology journalism chases stories of exceptional success. Reading tributes to the book by engineers, I am struck by how often they note triumphs and tragedies in their own careers that parallel those experienced by Kidder's characters. One was hit by a visceral sense of “grim familiarity” when he encountered a passing reference to the killing of a beloved project during a “big shootout at HoJo's.” “If you haven't yet had your own shoot-out at HoJo's,” he warned, “it is regrettably coming; may your career be blessed with few such firefights!”² Perhaps Kidder was really describing himself when he noted that West “was always finding romance and excitement in the seemingly ordinary.” His next book, after all, discovered equal wonder in the building of a single house.

9: It Makes Engineers Seem Pure and Noble

Kidder invokes Victorian critic John Ruskin's romantic idea that in building Europe's great cathedrals, ancient craftsmen experienced "the sort of work that gave meaning to life." According to Kidder, the engineers likewise "did the work, both with uncommon spirit and for reasons that, in a most frankly commercial setting, seemed remarkably pure." None of Kidder's characters become spectacularly rich or expected to, though they had hoped vainly for some financial recognition. When the team's success went unrewarded with stock options or bonuses, Kidder likened the rewards of computer engineering to those of pinball: the only thing you can win is a chance to play the game again. Key members of West's team leave Data General after he is banished to Japan, looking to play their next game of pinball elsewhere.

Late in the book, a unionized technician drops his pay stub into a trash basket. A senior engineer thereby discovers that the technician is making twice his own pay, thanks to overtime. His supervisor burns the evidence, "so that the troops wouldn't see it." The sacrifices of the engineers seemed even purer in contrast with the sales manager whose declaration on the final page that humans are motivated by "ego and the money to buy things that they and their families want" reads like blasphemy. Kidder finishes the book with: "It was a different game now. Clearly the machine no longer belonged to its makers."

Computing was a sideshow 40 years ago. Today Apple, Amazon, Microsoft, and Alphabet are the first trillion-dollar companies. Technology and money are inseparable. Instead of pinball, a technology career is more like a slot machine where the goal is to pull on the handle repeatedly and hope to win a financial jackpot. Intel routinely granted stock options to engineers. Adopted by software firms like Microsoft, this became the standard way of luring engineering talent to successful companies, creating millionaires in unprecedented numbers. Other developers, seeking a longer shot at greater wealth, sought stakes in the startups that dominant firms increasingly treated as a source, via acquisition, of products and staff.

The founders and early investors in Data General got rich, but not the engineers who sustained its growth. Were they noble, or just exploited? Kidder may have romanticized the motivations of his characters. Twenty years ago, *Wired* magazine found most of them working in senior roles at startups. Some had become rich.⁶ Yet I am myself just romantic enough to fear that something important was lost. Tom West's melancholy pride at the end of the book is surely more representative of the experience of most development teams than the world-changing success and unimaginable riches that dominate the more familiar stories of Gates, Jobs, and Zuckerberg. In fact, many teams are disbanded before their work is done. Almost every component part of a software or hardware system is invisible to the world, the quality of its execution and elegance of its design known only to its creators. If a system passes into the world the quality of that work will be one of many factors deciding whether it thrives or is quickly forgotten. Systems are often doomed by bad marketing, undercapitalization, changing customer tastes, or an idea that was ahead of its moment. I hope that today's developers retain enough of the old ethic of pinball to find an intrinsic satisfaction in difficult work well done, so that they don't feel worthless if the industry eats their youth without paying out a financial jackpot.

10: It Is Beautifully Engineered

Above all, *Soul* is an extraordinarily well-crafted book. That means more than just well-turned sentences and snappy observations. Each finely tuned section fits smoothly into the structure of the book. Kidder's pacing is flawless, his character beats impeccably timed, and he manages to make a mass of contradictions seem like a faithful portrait of a complex world rather than a failure of craft. West remains unknowable and paradoxical. The engineers are both exploited and given an enviable opportunity for meaningful work. The project was a rebellion, tacitly orchestrated by senior managers.

Kidder's only previous book was, in his own estimation, a miserable failure. Kidder has described his 1970s self as "plainly ambitious" yet "young beyond his years." As a journeyman freelance magazine writer, he churned out words with "boundless energy" but

had no idea how to shape them into a publishable story. Richard Todd, his implausibly sympathetic editor at *The Atlantic Monthly* (and West's former college roommate) imposed many rounds of rejection and revision while Kidder fitfully learned his trade.

How, then, could Kidder suddenly produce a masterpiece? Kidder dedicated *Soul* to Todd, who undoubtedly deserved it. Yet Kidder's sudden growth as a writer owed something to his immersion in the culture of engineering. In 1982, newly laden with accolades, he appeared with West at the Computer Museum in Boston. "In some sense," Kidder explained, "writing a book is like building a computer."⁵ He had witnessed the profoundly creative nature of the design process, the aesthetic qualities of good engineering, and the pure joy of finding an elegant solution to a hard problem. As *Soul* took shape, Kidder spent a long confinement in Todd's office, spreading typewritten draft pages in piles on its floor to prune and rearrange them. Todd sometimes lifted prized passages to deliver the crushing news: "you could do without this." Kidder then "began to learn a skill ... how to fall out of love with my own words," so that he could eliminate good material that was "at odds with the whole."

West's team had likewise begun with architectural decisions, ruthlessly subsuming the parts to the whole. Studying the engineer's craft, Kidder had learned something about his own. His unique fusion of engineering and literature has outlasted the MV/8000 and the mini-computer industry. I expect it to outlast the PC and the smartphone too. ■

References

1. Brooks, F.P., Jr. *The Mythical Man Month: Essays on Software Engineering*. Addison-Wesley, Reading, MA, 1975.
2. Cantrill, B. Reflecting on *Soul of a New Machine*. The Observation Deck (blog), (Feb. 10, 2019); <https://bit.ly/35KX3fN>
3. Florman, S.C. *The Hardy Boys and the Microrikids Make a Computer*. *New York Times Book Review* (Aug. 23, 1981), 1.
4. Kidder, T. and Todd, R. *Good Prose: The Art of Nonfiction*. Random House, NY, 2013.
5. Kidder, T. and West, T. Inside 'The Soul of a New Machine'. *The Computer Museum Report* (Spring 1983), 5–8.
6. Ratcliffe, E. O. Engineers! *Wired* (Dec. 2000), 356–367.
7. Yourdon, E. *Death March: The Complete Software Developer's Guide to Surviving "Mission Impossible" Projects*. Prentice Hall, 1997.

Thomas Haigh (thomas.haigh@gmail.com) is a professor of history at the University of Wisconsin—Milwaukee and a Comenius visiting professor at Siegen University. His next book, *A New History of Modern Computing* will appear with MIT Press later this year.

Copyright held by author.