

## CHAPTER THIRTY

# Compiling the Voice of a Team

*Andy Oram*

**COMPANIES COME AND GO. THE LASTING ASSET IS THE PEOPLE WHO WORKED FOR THEM. PEOPLE** bring to their new jobs the sum of their experiences—not just technical skills, but interpersonal patterns that they build up through years of dealing with colleagues and managers.

In this light, we can balance the importance of a project's formal, stated deliverables against the seemingly peripheral experiences of its staff along the way. I'd bet that you could cite very few projects from your own career whose deliverables are still in use. Taking this observation into account, I've always felt that a project's impact on the growth of its staff is just as important as the defined project goal. In fact, I see that as the premise behind this book.

This principle carries the most force in extreme cases where projects are headed inexorably for failure. I had the misfortune of working on just such a project 20 years ago. The only positive result I can remember was a moment of illumination during which I pulled together the collective wisdom of the contributors and gave it a voice.

Personal survival becomes a triumph on this kind of project, because the project challenges your professional judgment on a daily basis: the need to square jerry-rigged solutions with the team's long-term responsibilities; to maintain your integrity and your

friends under unbearable scheduling pressures; to hear and nod at requests that both the bearer and the recipient know to be unfeasible, then take them back to your cubicle and adjust them to engineering realities.

Projects that threaten survival also expose the tender boundary where the ego—which is responsible for civilized behavior—threatens to dissolve back into the frenzies of the id. I hit this boundary once during the project described in this chapter. But an intrepid suggestion by my manager directed my anguish into a surprising recovery—giving a voice to the individual contributors on the project in a way I believe helped us all survive.

Looking back on our audacious stroke, I also see in it the seed of a new relationship between individual contributors and management. My manager and I were groping toward a view of corporate behavior that is currently hawked by leading management consultants in business journals and books facing out on bookstore shelves. We were yearning for a flattened, less vertical command structure based on direct communication up and down the hierarchy, and were exploring the power of digital technology to implement that structure.

In short, through an impulsive gesture, I anticipated a vision that would be articulated by organizational experts two decades later.

## A Gem from the Computing Past

Close your eyes and travel back—using your memory if you have been in the computer industry long enough, and your imagination if you have not—to a simpler and more innocent time when a small computer company could create and market a complete system from the ground up. In the 1970s and 1980s, with computer chips and other components as building blocks and a total staff of just a few hundred people, a company I will call Edom Engineering manufactured Unix systems with unusually powerful data processing capabilities. It could be considered a supplier of low-cost systems meeting high-end needs, and was very popular with small scientific and engineering labs whose research aspirations outpaced their budgets.

The eccentric personalities at Edom Engineering, many of whom you'll encounter during this story, made work constantly stimulating and sometimes pleasurable. I treasure the people there, with whom I shared so many critical moments. Many of them are likely to read this story and recognize the incident I'll describe. I hope they forgive me if their recollection of the events differs from mine; I have done my best to reproduce the feelings and facts of a time that is more than two decades removed from now.

Edom Engineering occupied a low-slung warehouse-like facility on Boston's Other Technology Highway (Route 495, as opposed to the more famous Route 128). The bottom floor held a manufacturing facility and the top floor contained the engineering teams. The company was a class society, but a harmonious and very efficient one. A range of hardware from motherboards to data acquisition devices were created from scratch and tightly bound to specialized operating system software, compilers, and other tools.

Not many teams nowadays hold the kinds of conversations I sat in on, where compiler writers discuss bugs in register allocation heuristics and how to squeeze an extra bit of precision from a square root function. The engineers at Edom Engineering, including a high percentage of doctorates, were a smart bunch with a wonderful esprit de corps.

Our compiler group was tasked with reproducing all the features of the classic, industry-dominant FORTRAN compiler that Digital Equipment Corporation produced for its VAX line. One engineer got so carried away with this mission that she discovered the algorithm used by Digital's random number generator (actually, of course, a pseudorandom number generator) and made sure our RAND function produced the same sequence of numbers as the Digital library's RAND function.

I remember asking her, "Did the project requirements really make it necessary for you to produce the same exact random numbers?"

She shrugged and answered, "I just figured it out, and decided I might as well make our library conform as much as possible."

It was just such a combination of flamboyant showmanship, competitive zeal, and the sheer fun of creation that led Edom Engineering staff to retool the Unix operating system and all other software components to maximize the effectiveness of the company's well-regarded hardware. Our marketing staff boasted that we had the industry's broadest support for a wide range of features in both System V and BSD Unix, the two fundamental sources for Unix systems of that time.

Projects were organized around a cross-disciplinary model that became popular in the 1970s, largely influenced by Digital's adoption of matrix management. Engineers on each team interacted intimately with the marketing person, tech support staff, and technical writers who worked on the project. These contributors attended all project meetings and checked in with each other on almost an hourly basis.

The company's concern for good manuals and respect for documentation staff led me to join the firm as a technical writer. I was associated most closely with the compiler team, but enjoyed the chance to work on a variety of projects.

And indeed, Edom Engineering's manuals were known throughout the industry for the same high quality as the rest of the company's offerings. Programmers could actually understand our manual on device drivers, and could produce a working driver after reading it. My first year was devoted to documenting the entire FORTRAN language, which looks like an exercise in redundancy but was actually necessary because we had bulked up standard FORTRAN with lots of VAX extensions and our own enhancements.

Our management style was also fairly relaxed and democratic for those times. We held on to a scruffy start-up culture even as our staff grew past 300. But this grounding in openness harbored a tectonic fault: top managers were closer to their investors than their employees. A recent merger had lengthened the distance between investors and line staff even more. This distance caused chronic grumbling and unkind jokes among the lower

echelons of the company. But for important engineering decisions, the managers and investors were always smart enough to listen to their preeminent technical staff.

In Edom Engineering's heyday, it seemed there was nothing our engineers couldn't do. When the thread paradigm established itself as an important element of programming, the compiler team added standard pthreads support to the C compiler in just a few months, and the graphics group was well on its way to making calls thread-safe in the gigantic Motif library (the leading graphics library on Unix at that time) when their project fell victim to the industry crunch that the company ran into during the late 1980s.

It is this economic crisis, and the self-destructive reaction taken by management, to which I now turn. Our industry suffered the kind of change that is grievously disruptive but by no means rare—the kind of paradigm shift that my readers will probably go through several times in their careers, no matter what field they work in—so this change deserves a few paragraphs to help you understand how limited were the options left to us.

## Rewiring

Throughout the 1980s, requirements rose while competition intensified in our segment of the engineering field, which can broadly be called scientific data processing. As hardware got faster, generic workstations became better at meeting customer requirements without need for the clever tricks we used at Edom Engineering to achieve maximum performance. Meanwhile, the standard libraries and advanced features of Unix operating systems grew to the point where even our crack programming staff couldn't reproduce everything customers expected.

Our core scientific and engineering market was also diminishing because of an unrelated external factor: the U.S. military in the 1980s reduced the research funding upon which many of our customers depended.

It was in the late 1980s when Edom Engineering managers decided on a leap that they hoped would establish a new beachhead in advanced computing. Throughout its existence, we had been happy basing our systems on Motorola chips. But the new wave of Reduced Instruction Set Computing (RISC) processors promised much better performance. The sages among our hardware engineering staff checked out these processors and selected one that they said could deliver the performance we needed.

The impetus behind RISC processors was the increasingly multilayered decision making required within conventional chips from Intel and Motorola, which computer scientists now categorized with the demeaning term *Complex Instruction Set Computing* (CISC). Conventional chips got that name because they contained large numbers of instructions, some narrowly tailored to particular software tasks. Different instructions sometimes require different numbers of cycles, forcing the chip's engineers on the hardware side, as well as compiler developers on the software side, to build in more complex scheduling. The weight of all these components led to a need for more wires connecting the processing unit to memory.

By the 1980s, computer scientists determined that increased overhead from these trends was eating up processor power, and concluded that they could create a competitive new generation of chips using a stripped-down set of very short, simple instructions that behaved in rigorously similar ways in a fixed number of cycles. Implementations from various companies hit the market in that decade. RISC became the major story in the trade press, and benchmarks bore out the inventors' predictions.

Thus began the Longjump project at Edom Engineering. It was fraught with uncertainty from the start. We were basing a system on a new product based on a new computing paradigm, and depending on a small firm with an unknown track record. At the same time as the hardware team wrapped the chip into an audacious new system trying to take advantage of every possible feature for speed, the software teams had to port our unique, finely tuned operating system, compiler, and libraries. Because RISC was so different an architecture from the Motorola chips we had always used, we had to base our compiler and a lot of our system utilities on those provided by the RISC vendor. The porting effort steered through uncharted waters.

An engineer I'll call George was appointed project leader for Longjump. George was a relatively young project leader, an earnest, restless fellow with an unusual personality for an engineer in those days. He bore a trim mustache, kept his shoulders slightly hunched as he sat in meetings, and looked about with tight lips that got tighter when he listened to news he didn't want to hear. He had a foot in marketing as well as engineering, and tended to wear suit jackets in an age when traditional computer nerd attire was even sloppier than it is today. He must have known that a more professional look would bolster his credibility when meeting with top management. And because this was his first attempt as project leader, he felt the need to evoke credibility.

At the company meeting where Longjump was presented, George stood before the assembled management and staff to make a brief speech that included a seemingly formulaic phrase: "I'd like to thank the company's managers for entrusting me with the leadership of this project."

I don't think he anticipated how the rest of us would react to this gracious statement. We all knew Longjump was a hazardous undertaking, and this was no moment for even a hint of timidity. By thanking management for their trust, he planted in our minds the seed of doubt that he could live up to it.

But ultimately, if George played any counterproductive role, it was by showing too much deference to his managers as they directed the project into stagnant channels.

## Coping

My philosophy, as stated at the beginning of this chapter, is both community-oriented and process-oriented. It's community-oriented because I believe every success and failure can be exploited to improve team building, and process-oriented because I believe every project decision should take into account the way it treats team members and resources.

In earlier, sunnier times, our technical writing group once held a meeting (similar to many such meetings held by Unix writers, I'm sure) about the reorganization of manpages, which are an ungainly collection of historical Unix documentation. After some forty-five minutes where we dissected the competing issues and established some precepts, I said, "The decision we eventually come up with is less important than the fact of our holding this conversation."

Some of the group were shocked by that luxurious approach to how we spent our time, but I believe it appealed to the technical writing manager, Alan. He talked regularly about bringing one's full self into the workplace, including one's emotional reactions and personal values. He once asked his staff to read a recently published management book whose central claim was that companies needed to be based on a vision in order to enjoy long-term success, and that this vision must pivot on some valuable contribution to customers, workers, or the larger society.

Alan designated me the team leader for the Longjump documentation—my first stint as a team leader, like George. I decided it was my job to protect my team as much as possible. We had no room to negotiate our workflows or deliverables. But I would try to save us from wasteful bureaucracy. Furthermore, I pledged to share with all the writers any information that could identify upcoming crises before they hit.

It turned out I had something to offer on the latter point. As team leader for the documentation, I attended meetings throughout the software side of the project. I took meticulous notes and filtered them down to the items that I thought would have an impact on my fellow writers. I made sure all the writers on the train were offered a view as far ahead on the railroad tracks as any of us.

The aspect of team management that most engineers approach with a groan is regular meetings. At such meetings, typically, staff drone on about their accomplishments for the week and go back to their cubicles no more informed than before. For the Longjump writers, I announced that I would post an empty agenda each week and that we would hold a meeting next week only if someone added an item to the agenda.

This small innovation signaled my striving for both efficiency and democracy. Writers thanked me for eliminating pointless meetings. However, any writer had the power to call a meeting simply by posting an agenda item.

In practice, I was the only person to post agenda items. When we did have meetings, I reported news of specific interest to the group, and we were usually finished in 15 minutes. We didn't use any cool Scrum-like techniques, but we hit on our own formula for maximizing the value of our time together.

## Coding

It's always useful to have a few tools to support changes in organizational behavior. As a modest example, I implemented my agenda-driven meeting schedule by posting a file in a shared directory. The likelihood of two people editing the file at the same time was negli-

gible, but the concepts of operating systems and threads had been drilled into me and I therefore felt a heightened sensitivity toward race conditions. Consequently, I bundled access to the agenda in a simple script that used some basic command such as `chmod` to provide a simple locking mechanism using the filesystem.

A bigger tools challenge was posed by the manpages I mentioned earlier. All of our language tools were based on a new compiler suite that came from our vendor. We decided that editing our manpages to reflect the new system would be prohibitive, so the decree came down from somewhere that we should convert the vendor's manpages to fit our system.

The problem was that the vendor had given us pages only in output format, and to accommodate our customers' needs we required the pages in source format. The typical output we had looked like this:

```
O^HO^HOP^HP^HPT^HT^HTI^HI^HIO^HO^HON^HN^HNS^HS^HS
```

```
-^H-^H-o^Ho^Ho _^Hf_^Hi_^Hl_^He
Output binary image to _^Hf_^Hi_^Hl_^He.
```

That's a brief description of the classic `-o` compiler option. What appears as `^H` here is actually a backspace character (ASCII value 8) that causes the terminal to superimpose the following character on the preceding one. Superimposing the same character three times, in the format shown, causes it to appear in bold on the terminal. Superimposing a character on an underscore produces underlining.

What we needed instead were the original troff macros used to generate manpages, in which the preceding sequence would look like this:

```
.SS Options
.TP
/fB-o/fP /fI-file/fP
Output binary image to /fI-file/fP.
```

How could a writer convert dozens of pages documenting hundreds of options from output to source format? Typically for the Longjump project, the human and organizational costs—the veritable absurdity of such a job—were ignored by management, and a junior writer, Kimberly, was tasked with doing the conversion by hand.

By the time news of this job reached me, Kimberly was beside herself. She was in her first year of the job, which was her crucial first year in the computer field, and was the type of new hire who was eager to demonstrate her industriousness and loyalty. She would go to any length to satisfy a request from management. But as we sat together scoping out the job, paging through screen after screen of garbage at her terminal, she cried out, "I don't see how I can ever finish this job. I don't even know where to start."

Why didn't the company go back to our vendor and insist they give us the manpages in source format? I don't know. Perhaps my manager and I just didn't have enough experience to argue forcefully enough for this action. Perhaps no one had noticed the problem before signing the contract, and there was no recourse now. Or perhaps the vendor sensed

that Edom Engineering posed a competitive threat, and maliciously withheld the source files. In any case, we seemed stuck.

I decided to apply my modest programming skills to bring the project within a human being's purview. The hoary old troff format, particularly in output form, resisted mechanized processing. Even advanced AI techniques would probably not suffice to recognize all elements of a page, such as the section heading (.SS) and tagged item paragraph (.TP).

But I found a woman/machine collaboration that worked, and wrote some Emacs Lisp macros to automate as much activity as possible. Kimberly needed only to find a familiar element—such as a section heading or tagged item—position the Emacs cursor (point) over it, and press a single key to convert a paragraph to the right format. When I showed her the basic keystrokes, she nearly jumped out of her seat in excitement, and her relief was so great that she launched into the conversion of her first manpage with actual pleasure. Later she told me, "Those macros saved my life."

## Capitulating

War correspondents in every conflict wire home stories of chauffeurs who try to avoid known roadblocks by taking detours through back country. Often the new route proves more deadly than the one they avoided. Such was the case with Longjump.

Speed on this project only caused time to accelerate, rather than to slow down as Einstein said it would. After only a month or two, our engineers realized that our new system components were not working together, and that software porting was going badly. We had no time to lose; we could hemorrhage at any roadblock. The investors insisted on meeting our schedule, so someone at the top laid down a radical shift of direction.

Instead of incorporating the RISC chip into a custom-designed and custom-built system, we would become a value-added retailer for another successful computer firm that was selling a computer based on that chip. Adopting their computer system would require only some scrambling around at the top of the device and software stack, rather than the intensive mixing and baking that the chefs of our engineering staff had originally planned.

Many of us felt our hearts sink upon hearing this strategy. Although it was certainly a blow to our pride to be using another vendor's product, we had substantive reasons for fearing the switch as well. Without our secret sauce, our craftsmanlike reengineering of the system from bottom to top, we would lose most of our performance advantages. We were, in effect, offering customers a generic computer system that happened to be backwardly compatible with our previous systems. Only the new RISC architecture would (hopefully) provide enough of a performance boost to make the move worthwhile.

Grumbling was evident throughout the halls in those days. Staff at lower levels sensed that George was not playing the role a project leader should play, which would entail explaining to management the full consequences of their choice.



A popular cubicle decoration at that period was a fanciful series of brief reports moving up the management chain from an individual contributor to a company CEO. The individual contributor labeled his project a stinking pile of manure. This crass assessment softened in each report as it rose through the management hierarchy. By the time the news reached the CEO, he was enthusiastically announcing the project's potential for growing sweet-smelling flowers.

I mentioned at the beginning of this article that to survive projects like Longjump, staff must constantly reconcile short-term tasks with their professional judgment. A couple of months into the project, the effort of such reconciliation became exhausting. Not every request could be solved by solutions such as the Emacs macros I gave Kimberly.

Engineers started prefacing or ending their task descriptions with comments like "Not that this will do any good" and "I don't see how this will work." But soon the comments stopped. Saying them to cubicle neighbors was redundant, because they all were in agreement already. But the comments were also seen as a waste of time, because top management was responding only to investor pressure.

Worst of all, none of the individual contributors could offer a better way to meet our goals and deadlines. But being engineers rather than investors, we felt we'd have more chance of success by lengthening the project schedule and creating a system that maintained our unique advantages than by hoodwinking our customers with a "me too" product whose resemblance to our earlier products was only skin-deep. The moment for such an argument had passed, however. No longer did the engineers' opinions carry weight in the company's major engineering decisions, as they had throughout its history.

## The Break

Attending virtually all the meetings on the Longjump project, I was in a position to hear the full range of the engineers' lamentations. After many weeks of sitting at vinyl-laminated tables in the blank-walled boxes of conference rooms, I felt overwhelmed by the burden of what I was taking in.

One day I broke it all to Alan. I expected a conventional "Buck up and carry on" speech with a sprinkle of praise and commiseration from him, but Alan's response astonished me entirely.

"I think," he told me in his deliberate speaking style that shaped and placed each word with discernment, "you should write up exactly what you told me and send it as an email to the entire company."

What a bizarre idea! Admittedly, in Edom Engineering's loose corporate culture, staff always felt comfortable using the companywide alias, which would instantly reach everybody from the manufacturing team to the CEO. The alias was used not only for official business, but to announce parties, reschedule soccer games, and exchange jokes. There was nothing sacred about the companywide alias.

But one could ask whether I had anything to say that deserved taking up people's time. Everything I told Alan was circulating freely in the halls. As I have already indicated, engineers stopped talking about these issues because they had held the conversation many times and felt it was over and done with.

Yet the thought of writing up my thoughts thrilled me. Every individual contributor felt the same away about Longjump, but each one felt it in isolation. They could not share feelings in any encounter group or therapy session (despite the provision of a corporate psychologist who did nothing). Living what we felt to be a lie, day by day, we had lost the tiny hold on sanity that our earlier grasp of the truth had left to us and were plummeting down a dark hole lined with impossible demands.

So it lifted my spirits to hope that all our frustrations and fears could be uttered once more. I think Alan could tell I was the one to put them into words. Not only was I a writer; my team leader responsibilities placed me at a little-noticed but crucial fulcrum in the exchange of information about the project.

I remember him expounding on his idea with his legs crossed casually and a straight-backed posture. He urged me not to pull my punches. "You laid out the problems that the teams faced very clearly when explaining them to me, and I encourage you strongly to be just as candid, just as direct, and just as uncompromising when you write it up for the company."

The course seemed dangerous, but by this time, any fear had been beaten out of me. Little thought of career repercussions entered my mind. I knew I had been appointed team leader because my work was respected, and in the middle of the crunch on this project, management couldn't afford to lose me or discipline me in any way. It was for Alan's safety more than my own that I asked him, "What will you say if top management complains?"

"Don't worry, I'll back you up completely, and I can't see any way management could take negative action." Delivered in the same modulated tone, Alan's words gave me complete confidence.

If I preserved the email I sent that afternoon, it must be on some format that few computer systems can read today, so I'm sorry to say I can't quote it directly. It was only a few paragraphs long. I wrote of a consensus among Longjump team members that the project had taken a wrong turn. I summarized the tasks remaining to be done over the next few weeks and the hurdles that various teams faced with performance issues, lack of knowledge about the platform, and incompatibilities. I then turned to the effect of our observations on team morale, ending with our love for our work, our customers, and our company, all of which we thought would be jeopardized by forcing us to stick to the current plan and schedule.

I don't believe much work got done in the facility the rest of the day. Printouts of my email went up on cubicle walls. People congratulated me in the hall with comments such as, "I showed this to my wife and she asked how long it took before the person who wrote

it got fired.” Emails shot back from a sister facility in another state, where I was not well known, as people asked who this courageous and perspicacious writer was.

In fact, I don’t feel this is at all a story about courage or perspicacity. I was not courageous because I had no expectation of harm. I was not unusually perspicacious because I merely reported what I heard from everyone, and I presented my observations from that standpoint.

What I do feel proud of—and thank Alan for pushing me to do—is to have given my colleagues a voice. In doing this I broke through each individual’s feeling of isolated panic, and fought the virulent sense of helplessness pervading the company. In giving them a voice, I allowed them to survive.

## Anticipating Twenty-First Century Management

It was only while writing up my recollections as a chapter for *Beautiful Teams* that I reckoned there was something deeper and more significant about my email blast than mere letting off steam. I think Alan and I were exploring a new management style and the role of a new communications technology. The significance wouldn’t emerge for 20 years, until management consultants dubbed it a new trend in corporate behavior.

I pointed out at the beginning of this chapter that Edom Engineering management maintained a respect for engineers and a tolerance for their opinionated way of expressing themselves, a legacy of the company’s start-up days. Something on Longjump caused this to break down—probably the reluctance of the CEO to argue with investors and for George to argue with the CEO, along with the new lines of command brought in by our merger. The engineers at the grass roots were not prepared to organize themselves to preserve their decision-making power in the face of this breakdown in corporate culture. But another force, technological in nature, stepped in to offer an alternative power arrangement.

We were living in a period before mass online participation, when the World Wide Web ran on only a few dozen sites. Yet already, online communities had experimented with a grass-roots political activity that dispensed with traditional leaders and party centers. Although Howard Rheingold’s influential book *The Virtual Community: Homesteading on the Electronic Frontier* was not published until 1994,\* and John Perry Barlow didn’t release his Declaration of the Independence of Cyberspace (now seen as something of a period piece) until 1996, knowledge of Rheingold’s WELL was widespread among Internet users at the time I worked on Longjump, and I was already building a different way for people to work together with far-flung colleagues through email forums run by Computer Professionals for Social Responsibility.

I am convinced that Alan’s challenge to me to send out my views through email was a response to the power he saw dormant in digital communications. Alan does not remember this time in his career well enough to say whether he was consciously working

\* Alan was later to publish a critique of this book, but that doesn’t detract from my observation that he was making use of online communication and noticing much of its potential.

through its potential as I babbled away about deadlines and bug reports in his office, but the effect of our conspiracy was to let loose into the environment an unscheduled experiment in grass-roots participation. My immediate enthusiasm showed that I, too, prepped by my political work online, was equipped to create the experimental organism we unleashed.

The results of the experiment validated the premise. Digital networks are powerful enablers for democratic action, at least among groups of people that already have ties. In such a degenerate corporate environment, so constrained financially, we couldn't expect miraculous results. The observed effects on employee morale were the best outcome we could hope for. But one could easily see how the same use of technology could actually support constructive project management in a healthier company, if deployed at any earlier stage and supported by enlightened management and social norms.

The vision that Alan and I presented that day—unconsciously in my case, but perhaps consciously in his—now fills the pages of management texts. It has become, in the wake of the *Cluetrain Manifesto*, Eric Von Hippel's *Sources of Innovation*, and other books expounding on the value of openness, a hallmark of enlightened management. Few companies have moved yet to tap so radically the potential of employees (or customers and other stakeholders) organized in these ways into articulate policymakers for the organization. But many companies are talking the talk, and eventually that behavior will be accepted as normal.

## Final Notes

Edom Engineering staff weren't the types to wait around and let algae gather on the pond. On the day of my email, they read and responded to it, but then plunged back into the depths of their work.

No manager called on me to justify or explain my email—they probably knew very well what was going on, whether or not George had been forthright with them—but the direction of the Longjump project did not change, either. Ultimately, with the new business model based on reselling a competitor's system, we met our stated goals and finished the project on time with a formal success status. The product fizzled in the marketplace.

As a matter of fact, RISC chips in general never lived up to the hype. Conventional chip makers found ways to improve speed despite the supposedly crippling complexity of their designs, using such strategies as prefetching instructions, predictive scheduling, and multi-cores. Some of these strategies, such as breaking large instructions into smaller parts with a consistent structure, they borrowed from RISC design.

The conventional manufacturers kept costs low through economies of scale, and invested the profits they garnered through market dominance in high-priced, state-of-the-art facilities that could produce chips even more cheaply. RISC manufacturers never caught up. So although RISC companies remain in the field, RISC as a new computer market turned out to be a miscalculation by a large group of computer industry pundits and investors. Edom Engineering's last, great hope turned out to be just a footnote in that story.

I was never fired. I stayed at the company a few more months, until the remaining project of interest that I was documenting—the thread-safe Motif library—got canceled.

At that point, I went to my manager's manager, who headed a large engineering team and had an excellent rapport with me. I said to him, "My major project just got canceled, and I don't have enough work—what would you like me to do?"

This, of course, was a code phrase for: *I don't see any future for myself in this company, and I need your reassurance to stay.*

My manager's manager smiled in his perennially congenial way and answered, "Don't worry about it. Take some time to do some studying."

I took that as a code phrase for: *I have nothing to offer you. Go ahead and look for a new job.*

My departure was part of a general exodus during the company's subsequent and vertiginous decline. Since then, I have barely followed the sequence of acquisitions by other lackluster companies in the field. I feel now that it would have been better for everyone—the employees, the investors, and the customers—if the exodus had occurred before the Longjump project; it did no one a favor to push for the preservation of a dying business model.

But because we all chose (out of our affection for the products on which we had worked so hard during the years) to participate in this last gasp of a company we had built, we could at least benefit from the models we developed for behaving under extreme business conditions. I can't claim that my email blast was the proper way to channel the anger and fear that I was absorbing from those around me, but I aver that it was a psychological survival tactic that helped many of my colleagues. More interestingly, it was a somewhat distorted precursor of open communications on modern digital networks that underlie the more bottom-up, contributor-driven, democratic corporation being promoted by business experts today.

