CHAPTER SEVEN

AUTOMATION FOR THE PEOPLE

WHO NEEDS HUMANS, ANYWAY?

That question, in one rhetorical form or another, comes up frequently in discussions of automation. If computers are advancing so rapidly, and if people by comparison seem slow, clumsy, and error prone(fallible), why not build immaculately(perfectly) self-contained systems that perform flawlessly without any human oversight or intervention? Why not take the human factor out of the equation altogether? "We need to let robots take over," declared the technology theorist Kevin Kelly in a 2013 *Wired* cover story. He pointed to aviation as an example: "A computerized brain known as the autopilot can fly a 787 jet unaided, but irrationally(unreasonably) we place human pilots in the cockpit to babysit the autopilot 'just in case.'" [1] The news that a person was driving the Google car that crashed in 2011 prompted a writer at a prominent technology blog to exclaim, "More robo-drivers!"[2] Commenting on the struggles of Chicago's public schools, *Wall Street Journal* writer Andy Kessler remarked, only half-jokingly, "Why not forget the teachers and issue all 404,151 students an iPad or Android tablet?"[3] In a 2012 essay, the respected Silicon Valley venture capitalist Vinod Khosla suggested that health care will be much improved when medical software—which he dubs(name) "Doctor Algorithm"—goes from assisting primary-care physicians in making diagnoses to replacing the doctors entirely. "Eventually," he wrote, "we won't need the average doctor." [4] The cure for imperfect automation is total automation.

That's a seductive(attractive) idea, but it's simplistic. Machines share the fallibility(fallible; the likelihood of making errors) of their makers. Sooner or later, even the most advanced technology will break down, misfire, or, in the case of a computerized system, encounter a cluster of circumstances that its designers and programmers never anticipated and that leave its algorithms baffled(be too difficult to understand). In early 2009, just a few weeks before the Continental Connection crash in Buffalo, a US Airways Airbus A320 lost all engine power after hitting a flock of Canada geese on takeoff from LaGuardia Airport in New York. Acting quickly and coolly, Captain Chesley Sullenberger and his first officer, Jeffrey Skiles, managed, in three harrowing(frightening) minutes, to ditch(在海上迫降) the crippled jet safely in the Hudson River. All passengers and crew were evacuated(撤离). If the pilots hadn't been there to "babysit" the A320, a craft with state-of-the-art(most advanced) automation, the jet would have crashed and everyone on board would almost certainly have perished(die esp. from a disaster). For a passenger jet to have all its engines fail is rare. But it's not rare for pilots to rescue planes from mechanical malfunctions(失灵), autopilot glitches(bug), rough weather, and other unexpected events. "Again and again," Germany's *Der Spiegel* reported in a 2009 feature(特写) on airline safety, the pilots of fly-by-wire planes "run into new, nasty(worrisome & difficult) surprises that none of the engineers had predicted." [5]

The same is true elsewhere. The mishap(accident) that occurred while a person was driving Google's Prius was widely reported in the press; what we don't hear much about are all the times the backup drivers in Google cars, and other automated test vehicles, have to take the wheel to perform maneuvers the computers can't handle. Google requires that people drive its cars manually when on most urban and residential streets, and any employee who wants to operate one of the vehicles has to complete rigorous training in emergency driving techniques.[6] Driverless cars aren't quite as driverless as they seem.

In medicine, caregivers often have to overrule(推翻) misguided instructions or suggestions offered by clinical computers. Hospitals have found that while computerized drug-ordering systems alleviate(reduce) some common errors in dispensing medication, they introduce new problems. A 2011 study at one hospital revealed that the incidence of duplicated(复制的) medication orders actually increased after drug ordering was automated.[7] Diagnostic software is also far from perfect. Doctor Algorithm may well give you the right diagnosis and treatment most of the time, but if your particular set of symptoms doesn't fit the probability profile, you're going to be glad that Doctor Human was there in the examination room to review and overrule the computer's calculations.

As automation technologies become more complicated and more interconnected, with a welter of links and dependencies among software instructions, databases, network protocols, sensors, and mechanical parts, the potential sources of failure multiply. Systems become susceptible to what scientists call "cascading failures(级联故障)," in which a small malfunction in one component sets off a far-flung(widespread) and catastrophic chain of breakdowns. Ours is a world of "interdependent networks," a group of physicists reported in a 2010 *Nature* article. "Diverse infrastructures such as water supply, transportation, fuel and power stations are coupled together" through electronic and other links, which ends up making all of them "extremely sensitive to random failure." That's true even when the connections are limited to exchanges of data.[8]

Vulnerabilities become harder to discern(distinguish or identify) too. With the industrial machinery of the past, explains MIT computer scientist Nancy Leveson in her book *Engineering a Safer World,* "interactions among components could be thoroughly planned, understood, anticipated, and guarded against," and the overall design of a system could be tested exhaustively before it was put into everyday use. "Modern, high-tech systems no longer have these properties." They're less "intellectually manageable" than were their nuts-and-bolts(螺钉和螺母) predecessors.[9] All the parts may work flawlessly, but a small error or oversight(overlook) in system design—a glitch that might be buried in hundreds of thousands of lines of software code—can still cause a major accident.

The dangers are compounded(加重) by the incredible speed at which computers can make decisions and trigger actions. That was demonstrated over the course of a hair-raising(frightening) hour on the morning of August 1, 2012, when Wall Street's largest trading firm, Knight Capital Group, rolled out a new automated program for buying and selling shares. The cutting-edge(state-of-the-art/most advanced) software had a bug that went undetected during testing. The program immediately flooded exchanges with unauthorized and irrational orders, trading $2.6 million worth of stocks every second. In the forty-five minutes that passed before Knight's mathematicians and computer scientists were able to track the problem to its source and shut the offending program down, the software racked up(累计) $7 billion in errant(false) trades. The company ended up losing almost half a billion dollars, putting it on the verge of(濒临) bankruptcy. Within a week, a consortium(财团) of other Wall Street firms bailed Knight out(help out) to avoid yet another disaster in the financial industry.

Technology improves, of course, and bugs get fixed. Flawlessness, though, remains an ideal that can never be achieved. Even if a perfect automated system could be designed and built, it would still operate in an imperfect world. Autonomous cars don't drive the streets of utopia. Robots don't ply their trades in Elysian(天堂的) factories. Geese flock. Lightning strikes. The conviction that we can build an entirely self-sufficient, entirely reliable automated system is itself a manifestation of automation bias.

Unfortunately, that conviction is common not only among technology pundits(authority) but also among engineers and software programmers— the very people who design the systems. In a classic 1983 article in the journal *Automatica,* Lisanne Bainbridge, an engineering psychologist at University College London, described a conundrum(难题) that lies at the core of computer automation. Because designers often assume that human beings are, unreliable and inefficient at least when compared to a computer, they strive to give them as small a role as possible in the operation of systems. People end up functioning as mere monitors, passive watchers of screens.[10] That's a job that humans, with our notoriously wandering minds, are particularly bad at. Research on vigilance, dating back to studies of British radar operators watching for German submarines during World War II, shows that even highly motivated people can't keep their attention focused on a display of relatively stable information for more than about half an hour,[11] They get bored; they daydream; their concentration drifts. "This means," Bainbridge wrote, "that it is humanly impossible to carry out the basic function of monitoring for unlikely abnormalities.""

And because a person's skills "deteriorate when they are not used," she added, even an experienced system operator will eventually begin to act like "an inexperienced one" if his main job consists of watching rather than acting. As his instincts and reflexes grow rusty from disuse, he'll have trouble spotting and diagnosing problems, and his responses will be slow and deliberate(slow & careful) rather than quick and automatic. Combined with the loss of situational awareness, the degradation of know-how raises the odds that when something goes wrong, as it sooner or later will, the operator will react ineptly(unskillful; clumsy). And once that happens, system designers will work to place even greater limits on the operator's role, taking him further out of the action and making it more likely that he'll mess up in the future. The assumption that the human being will be the weakest link in the system becomes self-fulfilling(自我应验的).

■■■■

ERGONOMICS(人体工程学)**, THE** art and science of fitting tools and workplaces to the people who use them, dates back at least to the Ancient Greeks. Hippocrates, in "On Things Relating to the Surgery," provides precise instructions for how operating rooms should be lit and furnished, how medical instruments should be arranged and handled, even how surgeons should dress. In the design of many Greek tools, we see evidence of an exquisite(delicate 细腻的) consideration of the ways an implement's(tool or instrument) form, weight, and balance affect a worker's productivity, stamina(耐力/持久力), and health. In early Asian civilizations, too, there are signs that the instruments of labor were carefully designed with the physical and psychological well-being of the worker in mind.[13]

It wasn't until the Second World War, though, that ergonomics began to emerge, together with its more theoretical cousin cybernetics(控制论), as a formal discipline. Many thousands of inexperienced soldiers and other recruits had to be entrusted with complicated and dangerous weapons and machinery, and there was little time for training. Awkward designs and confusing controls could no longer be tolerated. Thanks to trailblazing(pioneering) thinkers like Norbert Wiener and U.S. Air Force psychologists Paul Fitts and Alphonse Chapanis, military and industrial planners came to appreciate that human beings play as integral (essential) a role in the successful workings of a complex technological system as do the system's mechanical components and electronic regulators. You can't optimize(优化) a machine and then force the worker to adapt to it, in rigid Taylorist fashion; you have to design the machine to suit the worker.

Inspired at first by the war effort and then by the drive to incorporate computers into commerce, government, and science, a large and dedicated group of psychologists, physiologists, neurobiologists, engineers, sociologists, and designers began to devote their varied talents to studying the interactions of people and machines. Their focus may have been the battlefield and the factory, but their aspiration was deeply humanistic: to bring people and technology together in a productive, resilient, and safe symbiosis(共生关系), a harmonious human-machine partnership that would get the best from both sides. If ours is an age of complex systems, then ergonomists are our metaphysicians(研究形而上学的人).

At least they should be. All too often, discoveries and insights from the field of ergonomics, or, as its now commonly known, human-factors engineering, are ignored or given short shrift(negligence). Concerns about the effects of computers and other machines on people's minds and bodies have routinely been trumped(win over) by the desire to achieve maximum efficiency, speed, and precision——or simply to turn as big a profit as possible. Software programmers receive little or no training in ergonomics, and they remain largely oblivious to relevant human-factors research. It doesn't help that engineers and computer scientists, with their strict focus on math and logic, have a natural antipathy(dislike) toward the "softer" concerns of their counterparts in the human-factors field. A few years before his death in 2006, the ergonomics pioneer David Meister, recalling his own career, wrote that he and his colleagues "always worked against the odds so that anything that was accomplished was almost unexpected." The course of technological progress, he wistfully(longingly; unsatisfactorily) concluded, "is tied to the profit motive; consequently, it has little appreciation of the human."[14]

It wasn’t always so. People first began thinking about technological progress as a force in history in the latter half of the eighteenth century, when the scientific discoveries of the Enlightenment began to be translated into the practical machinery of the Industrial Revolution. That was also, and not coincidentally, a time of political upheaval(violent disturbance 动荡). The democratic, humanitarian ideals of the Enlightenment culminated in the revolutions in America and France, and those ideals also infused(灌输) society’s view of science and technology. Technical advances were valued—by intellectuals, if not always by workers—as means to political reform. Progress was defined in social terms, with technology playing a supporting role. Enlightenment thinkers such as Voltaire, Joseph Priestley, and Thomas Jefferson saw, in the words of the cultural historian Leo Marx, “the new sciences and technologies not as ends in themselves, but as instruments for carrying out a comprehensive transformation of society.”

By the middle of the nineteenth century, however, the reformist view had, at least in the United States, been eclipsed by (a loss of importance)a new and very different concept of progress in which technology itself played the starring role. "With the further development of industrial capitalism," writes Marx, “Americans celebrated the advance of science and technology with increasing fervor, but they began to detach(separate) the idea from the goal of social and political liberation." Instead, they embraced "the now familiar view that innovations in science-based technologies are in themselves a sufficient and reliable basis for progress.”[15] New technology, once valued as a means to a greater good(benefit or interest), came to be revered as a good in itself.

It’s hardly a surprise, then, that in our own time the capabilities of computers have, as Bainbridge suggested, determined the division of labor in complex automated systems. To boost productivity, reduce labor costs, and avoid human error—to further progress—you simply allocate control over as many activities as possible to software, and as software’s capabilities advance, you extend the scope of its authority even further. The more technology, the better. The flesh-and-blood operators are left with responsibility only for those tasks that the designers can’t figure out how to automate, such as watching for anomalies(anomaly 反常现象) or providing an emergency backup in the event of a system failure. People are pushed further and further out of what engineers term "the loop”—the cycle of action, feedback, and decision making that controls a system’s moment-by-moment operations.

Ergonomists call the prevailing approach *technology-centered automation.* Reflecting an almost religious faith in technology, and an equally fervent(intense) distrust of human beings, it substitutes misanthropic(厌恶人类的) goals for humanistic ones. It turns the glib(肤浅的) "who needs humans?" attitude of the technophilic(crazy about technology) dreamer into a design ethic. As the resulting machines and software tools make their way into workplaces and homes, they carry that misanthropic ideal into our lives. "Society," writes Donald Norman, a cognitive scientist and author of several influential books about product design, “has unwittingly(不知不觉间) fallen into a machine-centered orientation to life, one that emphasizes the needs of technology over those of people, thereby forcing people into a supporting role, one for which we are most unsuited. Worse, the machine-centered viewpoint compares people to machines and finds us wanting(lacking), incapable of precise, repetitive, accurate actions.” Although it now "pervades society," this view warps(skew;歪曲) our sense of ourselves. "It emphasizes tasks and activities that we should not be performing and ignores our primary skills and attributes—activities that are done poorly, if at all, by machines. When we take the machine-centered point of view, we judge things on artificial, mechanical merits."[16]

It's entirely logical that those with a mechanical bent(嗜好) would take a mechanical view of life. The impetus behind invention is often, as Norbert Wiener put it, "the desires of the gadgeteer(喜欢搞发明的人) to see the wheels go round."[17] And it's equally logical that such people would come to control the design and construction of the intricate systems and software programs that now govern or mediate(influence) society's workings. They're the ones who know the code. As society becomes ever more computerized, the programmer becomes its unacknowledged(not publicly recognized) legislator. By defining the human factor as a peripheral concern, the technologist also removes the main impediment(barrier) to the fulfillment of his desires; the unbridled(not controlled) pursuit of technological progress becomes self-justifying. To judge technology primarily on its technological merits is to give the gadgeteer carte blanche(全权委托).

In addition to fitting the dominant ideology of progress, the bias to let technology guide decisions about automation has practical advantages. It greatly simplifies the work of the system builders. Engineers and programmers need only take into account what computers and machines can do. That allows them to narrow their focus and winnow(遴选) a project's specifications. It relieves them of having to wrestle with the complexities, vagaries(vagary:不确定性), and frailties(frailty: weakness) of the human body and psyche. But however compelling(forceful) as a design tactic, the simplicity of technology-centered automation is a mirage(海市蜃楼). Ignoring the human factor does not remove the human factor.

In a much-cited 1997 paper, "Automation Surprises," the human-factors experts Nadine Sarter, David Woods, and Charles Bill-ings traced the origins of the technology-focused approach. They described how it grew out of and continues to reflect the "myths, false hopes, and misguided intentions associated with modern technology." The arrival of the computer, first as an analogue machine(模拟机) and then in its familiar digital form, encouraged engineers and industrialists to take an idealistic view of electronically controlled systems, to see them as a kind of cure-all for human inefficiency and fallibility. The order and cleanliness of computer operations and outputs seemed heaven-sent when contrasted with the earthly messiness of human affairs. "Automation technology," Sarter and her colleagues wrote, "was originally developed in hope of increasing the precision and economy of operations while, at the same time, reducing operator workload and training requirements. It was considered possible to create an autonomous system that required little if any human involvement and therefore reduced or eliminated the opportunity for human error." That belief led, again with pristine(original) logic, to the further assumption that "automated systems could be designed without much consideration for the human element in the overall system." [18]

The desires and beliefs underpinning(support) the dominant design approach, the authors continued, have proved naive and damaging. While automated systems have often enhanced the "precision and economy of operations," they have fallen short of expectations in other respects, and they have introduced a whole new set of problems. Most of the shortcomings stem from "the fact that even highly automated systems still require operator involvement and therefore communication and coordination between human and machine." But because the systems have been designed without sufficient regard for the people who operate them, their communication and coordination capabilities are feeble(weak). In consequence, the computerized systems lack the "complete knowledge" of the work and the "comprehensive access to the outside world" that only people can provide. "Automated systems do not know when to initiate communication with the human about their intentions and activities or when to request additional information from the human. They do not always provide adequate feedback to the human who, in turn, has difficulties tracking automation status and behavior and realizing there is a need to intervene to avoid undesirable actions by the automation." Many of the problems that bedevil(harry; frustrate) automated systems stem from "the failure to design human-machine interaction to exhibit the basic competencies of human-human interaction." [19]

Engineers and programmers compound the problems when they hide the workings of their creations from the operators, turning every system into an inscrutable(mysterious) black box. Normal human beings, the unstated assumption goes, don't have the smarts or the training to grasp the intricacies of a software program or robotic apparatus. If you tell them too much about the algorithms or procedures that govern its operations and decisions, you'll just confuse them or, worse yet, encourage them to tinker with(胡乱修补/摆弄) the system. It's safer to keep people in the dark. Here again, though, the attempt to avoid human errors by removing personal responsibility ends up making the errors more likely. An ignorant operator is a dangerous operator. As the University of Iowa human-factors professor John Lee explains, it's common for an automated system to use "control algorithms that are at odds with the control strategies and mental model of the person [operating it]." If the person doesn't understand those algorithms, there's no way she can "anticipate the actions and limits of the automation." The human and the machine, operating under conflicting assumptions, end up working at cross-purposes. People's inability to comprehend the machines they use can also undermine(make weaker gradually) their self-confidence, Lee reports, which "can make them less inclined to intervene" when something goes wrong.[20]

■■■■

HUMAN-FACTORS EXPERTS have long urged designers to move away from the technology-first approach and instead embrace *human-centered automation.* Rather than beginning with an assessment of the capabilities of the machine, human-centered design begins with a careful evaluation of the strengths and limitations of the people who will be operating or otherwise interacting with the machine. It brings technological development back to the humanistic principles that inspired the original ergonomists. The goal is to divide roles and responsibilities in a way that not only capitalizes on(draw on; use) the computers speed and precision but also keeps workers engaged, active, and alert—in the loop rather than out of it.[21]

Striking that kind of balance isn't hard. Decades of ergonomic research show it can be achieved in a number of straightforward ways. A system's software can be programmed to shift control over critical functions from the computer back to the operator at frequent but irregular intervals. Knowing that they may need to take command at any moment keeps people attentive and engaged, promoting situational awareness and learning. A design engineer can put limits on the scope of automation, making sure that people working with computers perform challenging tasks rather than being relegated(lower) to passive, observational roles. Giving people more to do helps sustain the generation effect. A designer can also give the operator direct sensory feedback on the system's performance, using audio and tactile alerts as well as visual displays, even for those activities that the computer is handling. Regular feedback heightens engagement and helps operators remain vigilant(alert/attentive).

One of the most intriguing applications of the human-centered approach is *adaptive automation.* In adaptive systems, the computer is programmed to pay close attention to the person operating it. The division of labor between the software and the human operator is adjusted continually, depending on what's happening at any given moment.[22] When the computer senses that the operator has to perform a tricky(thorny) maneuver, for example, it might take over all the other tasks. Freed from distractions, the operator can concentrate her full attention on the critical challenge. Under routine conditions, the computer might shift more tasks over to the operator, increasing her workload to ensure that she maintains her situational awareness and practices her skills. Putting the analytical capabilities of the computer to humanistic use, adaptive automation aims to keep the operator at the peak of the Yerkes-Dodson performance curve(可回看 *Interlude, with Dancing Mice*), preventing both cognitive overload and cognitive underload. DARPA, the Department of Defense laboratory that spearheaded(lead) the creation of the internet, is even working on developing "neuroergonomic" systems that, using various brain and body sensors, can "detect an individual's cognitive state and then manipulate task parameters to overcome perceptual, attentional, and working memory bottlenecks."[23] Adaptive automation also holds promise for injecting a dose of humanity into the working relationships between people and computers. Some early users of the systems report that they feel as though they're collaborating with a colleague rather than operating a machine.

Studies of automation have tended to focus on large, complex, and risk-laden systems, the kind used on flight decks, in control rooms, and on battlefields. When these systems fail, many lives and a great deal of money can be lost. But the research is also relevant to the design of decision-support applications used by doctors, lawyers, managers, and others in analytical trades. Such programs go through a lot of personal testing to make them easy to learn and operate, but once you dig beneath the user-friendly interface, you find that the technology-centered ethic still holds sway(dominate). "Typically," writes John Lee, "expert systems act as a prosthesis(假体), supposedly replacing flawed and inconsistent human reasoning with more precise computer algorithms."[24] They're intended to supplant(replace), rather than supplement(add), human judgment. With each upgrade in an application's data-crunching speed and predictive acumen(机敏), the programmer shifts more decision-making responsibility from the professional to the software.

Raja Parasuraman, who has studied the personal consequences of automation as deeply as anyone, believes this is the wrong approach. He argues that decision-support applications work best when they deliver pertinent information to professionals at the moment they need it, without recommending specific courses of action.[25] The smartest, most creative ideas come when people are afforded room to think. Lee agrees. "A less automated approach, which places the automation in the role of critiquing the operator, has met with much more success." he writes. The best expert systems present people with "alternative interpretations, hypotheses, or choices." The added and often unexpected information helps counteract the natural cognitive biases that sometimes skew human judgment. It pushes analysts and decision makers to look at problems from different perspectives and consider broader sets of options. But Lee stresses that the systems should leave the final verdict(decision) to the person. In the absence of perfect automation, he counsels, the evidence shows that "a lower level of automation, such as that used in the critiquing approach, is less likely to induce errors."[26] Computers do a superior job of sorting through lots of data quickly, but human experts remain subtler and wiser thinkers than their digital partners.

Carving out(open/develop) a protected space for the thoughts and judgments of expert practitioners is also a goal of those seeking a more humanistic approach to automation in the creative trades. Many designers criticize popular CAD programs for their pushiness. Ben Tranel, an architect with the Gensler firm in San Francisco, praises computers for expanding the possibilities of design. He points to the new, Gensler-designed Shanghai Tower in China, a spiraling, energy-efficient skyscraper, as an example of a building that "couldn't have been built" without computers. But he worries that the literalism(拘泥于字面，引申为刻板或限制发挥) of design software—the way it forces architects to define the meaning and use of every geometric element they input—is foreclosing(hamper; impede; stop) the open-ended, unstructured explorations that freehand sketching encouraged. "A drawn line can be many things," he says, whereas a digitized line has to be just one thing.[27]

Back in 1996, the architecture professors Mark Gross and Ellen Yi-Luen Do proposed an alternative to literal-minded(not imaginative) CAD software. They created a conceptual blueprint of an application with a "paper-like" interface that would be able to "capture users' intended ambiguity, vagueness, and imprecision and convey these qualities visually." It would lend design software "the suggestive power of the sketch."[28] Since then, many other scholars have made similar proposals. Recently, a team led by Yale computer scientist Julie Dorsey created a prototype of a design application that provides a "mental canvas." Rather than having the computer automatically translate two-dimensional drawings into three-dimensional virtual models, the system, which uses a touchscreen tablet as an input device, allows an architect to do rough sketches in three dimensions. ''Designers can draw and redraw lines without being bound by the constraints of a polygonal mesh(多边形网络) or the inflexibility of a parametric(参数的) pipeline," the team explained. "Our system allows easy iterative(repetitious) refinement throughout the development of an idea, without imposing geometric precision before the idea is ready for it."[29] With less pushy software, a designer's imagination has more chance to flourish.

■■■■

THE TENSION between technology-centered and human-centered automation is not just a theoretical concern of academics. It affects decisions made every day by business executives, engineers and programmers, and government regulators. In the aviation business, the two dominant airliner manufacturers have been on different sides of the design question since the introduction of fly-by-wire systems thirty years ago. Airbus pursues a technology-centered approach. Its goal is to make its planes essentially "pilot-proof.'' The company's decision to replace the bulky(large), front-mounted(前悬挂式的) control yokes that have traditionally steered planes with diminutive(mini-type), side-mounted joysticks was one expression of that goal. The game-like controllers send inputs to the flight computers efficiently, with minimal manual effort, but they don't provide pilots with tactile feedback. Consistent with the ideal of the glass cockpit, they emphasize the pilot's role as a computer operator rather than as an aviator. Airbus has also programmed its computers to override pilots' instructions in certain situations in order to keep the jet within the software-specified parameters of its flight envelope(飞行包线). The software, not the pilot, wields ultimate control.

Boeing has taken a more human-centered tack(approach; guideline) in designing its fly-by-wire craft. In a move that would have made the Wright brothers happy, the company decided that it wouldn't allow its flight software to override the pilot. The aviator retains final authority over maneuvers, even in extreme circumstances. And not only has Boeing kept the big yokes of yore(往昔); it has designed them to provide artificial feedback that mimics(imitate) what pilots felt back when they had direct control over a plane's steering mechanisms. Although the yokes are just sending electronic signals to computers, they've been programmed to provide resistance and other tactile cues that simulate the feel of the movements of the plane's ailerons(副翼), elevators(升降舵), and other control surfaces(控制键). Research has found that tactile, or haptic(触觉的), feedback is significantly more effective than visual cues alone in alerting pilots to important changes in a plane's orientation and operation, according to John Lee. And because the brain processes tactile signals in a different way than visual signals, "haptic warnings" don't tend to "interfere with the performance of concurrent visual tasks."[31] In a sense, the synthetic(虚拟的), tactile feedback takes Boeing pilots out of the glass cockpit. They may not wear their jumbo jets the way Wiley Post wore his little Lockheed Vega, but they are more involved in the bodily experience of flight than are their counterparts on Airbus flight decks.

Airbus makes magnificent planes. Some commercial pilots prefer them to Boeing's jets, and the safety records of the two manufacturers are pretty much identical. But recent incidents reveal the shortcomings of Airbus's technology-centered approach. Some aviation experts believe that the design of the Airbus cockpit played a part in the Air France disaster. The voice-recorder transcript(written record) revealed that the whole time the pilot controlling the plane, Pierre-Cedric Bonin, was pulling back on his sidestick, his copilot, David Robert, was oblivious to Bonin's fateful mistake. In a Boeing cockpit, each pilot has a clear view of the other pilot's yoke and how it's being handled. If that weren't enough, the two yokes operate as a single unit. If one pilot pulls back on his yoke, the other pilot's goes back too. Through both visual and haptic cues, the pilots stay in sync(保持同步). The Airbus sidesticks, in contrast, are not in clear view, they work with much subtler motions, and they operate independently. It's easy for a pilot to miss what his colleague is doing, particularly in emergencies when stress rises and focus narrows.

Had Robert seen and corrected Bonin's error early on, the pilots may well have regained control of the A330. The Air France crash, Chesley Sullenberger(萨利机长) has said, would have been "much less likely to happen" if the pilots had been flying in a Boeing cockpit with its human-centered controls.[32] Even Bernard Ziegler, the brilliant and proud French engineer who served as Airbus's top designer until his retirement in 1997, recently expressed misgivings about his company's design philosophy. "Sometimes I wonder if we made an airplane that is too easy to fly," he said to William Langewiesche, the writer, during an interview in Toulouse, where Airbus has its headquarters. "Because in a difficult airplane the crews may stay more alert." He went on to suggest that Airbus "should have built a kicker(auxiliary engine) into the pilots' seats."[33] He may have been joking, but his comment jibes with(be in line with) what human-factors researchers have learned about the maintenance of human skills and attentiveness. Sometimes a good kick, or its technological equivalent, is exactly what an automated system needs to give its operators.

When the FAA, in its 2013 safety alert for operators, suggested that airlines encourage pilots to assume manual control of their planes more frequently during flights, it was also taking a stand(take a position/stance), if a tentative one, in favor of human-centered automation. Keeping the pilot more firmly in the loop, the agency had come to realize, could reduce the chances of human error, temper(减轻) the consequences of automation failure, and make air travel even safer than it already is. More automation is not always the wisest choice. The FAA, which employs a large and respected group of human-factors researchers, is also paying close attention to ergonomics as it plans its ambitious "NextGen" overhaul(全面修改) of the nation's air-traffic-control system. One of the projects overarching(principal; dominant) goals is to "create aerospace systems that adapt to, compensate for, and augment(improve; increase) the performance of the human." [34]

In the financial industry, the Royal Bank of Canada is also going against the grain(be against the trend) of technology-centered automation. At its Wall Street trading desk, it has installed a proprietary software program, called THOR, that actually slows down the transmission of buy and sell orders in a way that protects them from the algorithmic manipulations of high-speed traders. By slowing the orders, RBC has found, trades often end up being executed at more attractive terms for its customers. The bank admits that it's making a tradeoff in resisting the prevailing technological imperative(order/rule) of speedy data flows. By eschewing(avoid) high-speed trading, it makes a little less money on each trade. But it believes that, over the long run, the strengthening of client loyalty and the reduction of risk will lead to higher profits overall.[35]

One former RBC executive, Brad Katsuyama, is going even further. Having watched stock markets become skewed(misguided) in favor of high-frequency traders, he spearheaded the creation of a new and fairer exchange, called IEX. Opened late in 2013, IEX imposes controls on automated systems. Its software manages the flow of data to ensure that all members of the exchange receive pricing and other information at the same time, neutralizing the advantages enjoyed by predatory(掠夺成性的) trading firms that situate their computers next door to exchanges. And IEX forbids certain kinds of trades and fee schemes that give an edge to speedy algorithms. Katsuyama and his colleagues are using sophisticated technology to level the playing field between people and computers. Some national regulatory agencies are also trying to put the brakes on automated trading, through laws and regulations. In 2012, France placed a small tax on stock trades, and Italy followed suit a year later. Because high-frequency-trading algorithms are usually designed to execute volume-based arbitrage(套利) strategies—each trade returns only a minuscule profit, but millions of trades are made in a matter of moments——even a tiny transaction tax can render the programs much less attractive.

■■■■

SUCH ATTEMPTS to rein in(约束) automation are encouraging. They show that at least some businesses and government agencies are willing to question the prevailing technology-first attitude. But these efforts remain exceptions to the rule, and their continued success is far from assured. Once technology-centered automation has taken hold in a field, it becomes very hard to alter the course of progress. The software comes to shape how work is done, how operations are organized, what consumers expect, and how profits are made. It becomes an economic and a social fixture. This process is an example of what the historian Thomas Hughes calls "technological momentum."[36] In its early development, a new technology is malleable(可改变的 adaptable); its form and use can be shaped not only by the desires of its designers but also by the concerns of those who use it and the interests of society as a whole. But once the technology becomes embedded in(inserted) physical infrastructure, commercial and economic arrangements, and personal and political norms and expectations, changing it becomes enormously difficult. The technology is at that point an integral component(整体构件) of the social status quo(现状). Having amassed great inertial force(惯性), it continues down the path it's on. Particular technological components will still become outdated, of course, but they'll tend to be replaced by new ones that refine and perpetuate the existing modes of operation and the related measures of performance and success.

The commercial aviation system, for example, now depends on the precision of computer control. Computers are better than pilots at plotting the most fuel-efficient routes, and computer-controlled planes can fly closer together than can planes operated by people. There's a fundamental tension between the desire to enhance pilots' manual flying skills and the pursuit of ever higher levels of automation in the skies. Airlines are unlikely to sacrifice profits and regulators are unlikely to curtail the capacity of the aviation system in order to give pilots significantly more time to practice flying by hand. The rare automation-related disaster, however horrifying, may be accepted as a cost of an efficient and profitable transport system. In health care, insurers and hospital companies, not to mention politicians, look to automation as a quick fix(速成法) to lower costs and boost productivity. They'll almost certainly keep ratcheting up(提高) the pressure on providers to automate medical practices and procedures in order to save money, even if doctors have worries about the long-term erosion of their most subtle and valuable talents. On financial exchanges, computers can execute a trade in ten microseconds—that's one ten-millionth of a second—but it takes the human brain nearly a quarter of a second to respond to an event or other stimulus. A computer can process tens of thousands of trades in the blink of a trader's eye.[37] The speed of the computer has taken the person out of the picture.

It's commonly assumed that any technology that comes to be broadly adopted in a field, and hence gains momentum, must be the best one for the job. Progress, in this view, is a quasi-Darwinian(类似达尔文进化论的) process. Many different technologies are invented, they compete for users and buyers, and after a period of rigorous testing and comparison the marketplace chooses the best of the bunch. Only the fittest tools survive. Society can thus be confident that the technologies it employs are the optimum ones—and that the alternatives discarded along the way were flawed in some fatal way. It's a reassuring view of progress, founded on, in the words of the late historian David Noble, "a simple faith in objective science, economic rationality, and the market." But as Noble went on to explain in his 1984 book *Forces of Production,* it's a distorted view: "It portrays technological development as an autonomous and neutral technical process, on the one hand, and a coldly rational and self-regulating process, on the other, neither of which accounts for people, power, institutions, competing values, or different dreams."[38] In place of the complexities, vagaries, and intrigues(conspiracy, scheme) of history, the prevailing view of technological progress presents us with a simplistic, retrospective fantasy.

Noble illustrated the tangled(complicated) way technologies actually gain acceptance and momentum through the story of the automation of the machine tool industry in the years after World War II. Inventors and engineers developed several different techniques for programming lathes(机床), drill presses(钻床), and other factory tools, and each of the control methods had advantages and disadvantages. One of the simplest and most ingenious of the systems, called Specialmatic, was invented by a Princeton-trained engineer named Felix P. Caruthers and marketed by a small New York company called Automation Specialties. Using an array of keys and dials to encode and control the workings of a machine, Specialmatic put the power of programming into the hands of skilled machinists on the factory floor. A machine operator, explained Noble, "could set and adjust feeds and speeds, relying upon accumulated experience with the sights, sounds, and smells of metal cutting."[39] In addition to bringing the tacit know-how of the experienced craftsman into the automated system, Specialmatic had an economic advantage: a manufacturer did not have to pay a squad of engineers and consultants to program its equipment. Caruthers's technology earned accolades(honor) from *American Machinist* magazine, which noted that Specialmatic "is designed to permit complete setup and programming at the machine." It would allow the machinist to gain the efficiency benefits of automation while retaining "full control of his machine throughout its entire machining cycle."[40]

But Specialmatic never gained a foothold (立足之处) in the market. While Caruthers was working on his invention, the U.S. Air Force was plowing money into a research program, conducted by an MIT team with long-standing ties to the military, to develop "numerical control," a digital coding technique that was a forerunner of modern software programming. Not only did numerical control(数字控制) enjoy the benefits of a generous government subsidy and a prestigious academic pedigree(birth); it appealed to business owners and managers who, faced with unremitting(constant) labor tensions, yearned to(desire) gain more control over the operation of machinery in order to undercut the power of workers and their unions. Numerical control also had the glow of a cutting-edge technology—it was carried along by the burgeoning (grow & flourish) postwar excitement over digital computers. The MIT system may have been, as the author of a Society of Manufacturing Engineers paper would later write, "a complicated, expensive monstrosity(terrible stuff)."[41] but industrial giants like GE and Westinghouse rushed to embrace the technology, never giving alternatives like Specialmatic a chance. Far from winning a tough evolutionary battle for survival, numerical control was declared the victor before competition even began. Programming took precedence over people, and the momentum behind the technology-first design philosophy grew. As for the general public, it never knew that a choice had been made.

Engineers and programmers shouldn't bear all the blame for the ill effects of technology-centered automation. They may be guilty at times of pursuing narrowly mechanistic dreams and desires, and they may be susceptible to the "technical arrogance" that "gives people an illusion of illimitable power," in the words of the physicist Freeman Dyson.[42] But they're also responding to the demands of employers and clients. Software developers always face a trade-off in writing programs for automating work. Taking the steps necessary to promote the development of expertise—restricting the scope of automation, giving a greater and more active role to people, encouraging the development of automaticity through rehearsal and repetition— entails a sacrifice of speed and yield. Learning requires inefficiency. Businesses, which seek to maximize productivity and profit, would rarely, if ever, accept such a trade-off. The main reason they invest in automation, after all, is to reduce labor costs and streamline operations.

As individuals, too, we almost always seek efficiency and convenience when we decide which software application or computing device to use. We pick the program or gadget that lightens our load, and frees up our time, not the one that makes us work harder and longer. Technology companies naturally cater to such desires when they design their wares. They compete fiercely to offer the product that requires the least effort and thought to use. "At Google and all these places," says Google executive Alan Eagle, explaining the guiding philosophy of many software and internet businesses, "we make technology as brain-dead easy to use as possible,"[43] When it comes to the development and use of commercial software, whether it underpins an industrial system or a smartphone app, abstract concerns about the fate of human talent can't compete with the prospect of saving time and money.

I asked Parasuraman whether he thinks society will come to use automation more wisely in the future, striking a better balance between computer calculation and personal judgment, between the pursuit of efficiency and the development of expertise. He paused a moment and then, with a wry laugh, said, "I'm not very sanguine."