

All Can Be Lost: The Risk of Putting Our Knowledge in the Hands of Machines

We rely on computers to fly our planes, find our cancers, design our buildings, audit our businesses. That's all well and good. But what happens when the computer fails?

By [Nicholas Carr](#)



Kyle Bean

[November 2013 Issue](#)

On the evening of February 12, 2009, a Continental Connection commuter flight made its way through blustery weather between Newark, New Jersey, and Buffalo, New York. As is typical of commercial flights today, the pilots didn't have all that much to do during the hour-long trip. The captain, Marvin Renslow, manned the controls briefly during takeoff, guiding the Bombardier Q400 turboprop into the air, then switched on the autopilot and let the software do the flying. He and his co-pilot, Rebecca Shaw, chatted—about their families, their careers, the personalities of air-traffic controllers—as the plane cruised uneventfully along its northwesterly route at 16,000 feet. The Q400 was well into its approach to the Buffalo airport, its landing gear down, its wing flaps out, when the pilot's control yoke began to shudder noisily, a signal that the plane was losing lift and risked going into an aerodynamic stall. The autopilot

disconnected, and the captain took over the controls. He reacted quickly, but he did precisely the wrong thing: he jerked back on the yoke, lifting the plane's nose and reducing its airspeed, instead of pushing the yoke forward to gain velocity. Rather than preventing a stall, Renslow's action caused one. The plane spun out of control, then plummeted. "We're down," the captain said, just before the Q400 slammed into a house in a Buffalo suburb.

The crash, which killed all 49 people on board as well as one person on the ground, should never have happened. A National Transportation Safety Board investigation concluded that the cause of the accident was pilot error. The captain's response to the stall warning, the investigators reported, "should have been automatic, but his improper flight control inputs were inconsistent with his training" and instead revealed "startle and confusion." An executive from the company that operated the flight, the regional carrier Colgan Air, admitted that the pilots seemed to lack "situational awareness" as the emergency unfolded.

The Buffalo crash was not an isolated incident. An eerily similar disaster, with far more casualties, occurred a few months later. On the night of May 31, an Air France Airbus A330 took off from Rio de Janeiro, bound for Paris. The jumbo jet ran into a storm over the Atlantic about three hours after takeoff. Its air-speed sensors, coated with ice, began giving faulty readings, causing the autopilot to disengage. Bewildered, the pilot flying the plane, Pierre-Cédric Bonin, yanked back on the stick. The plane rose and a stall warning sounded, but he continued to pull back heedlessly. As the plane climbed sharply, it lost velocity. The airspeed sensors began working again, providing the crew with accurate numbers. Yet Bonin continued to slow the plane. The jet stalled and began to fall. If he had simply let go of the control, the A330 would likely have righted itself. But he didn't. The plane dropped 35,000 feet in three minutes before hitting the ocean. All 228 passengers and crew members died.

The first automatic pilot, dubbed a "metal airman" in a 1930 *Popular Science* article, consisted of two gyroscopes, one mounted horizontally, the other vertically, that were connected to a plane's controls and powered by a wind-driven generator behind the propeller. The horizontal gyroscope kept the wings level, while the vertical one did the steering. Modern autopilot systems bear little resemblance to that rudimentary device. Controlled by onboard computers running immensely complex software, they gather

information from electronic sensors and continuously adjust a plane's attitude, speed, and bearings. Pilots today work inside what they call "glass cockpits." The old analog dials and gauges are mostly gone. They've been replaced by banks of digital displays. Automation has become so sophisticated that on a typical passenger flight, a human pilot holds the controls for a grand total of just three minutes. What pilots spend a lot of time doing is monitoring screens and keying in data. They've become, it's not much of an exaggeration to say, computer operators.

And that, many aviation and automation experts have concluded, is a problem. Overuse of automation erodes pilots' expertise and dulls their reflexes, leading to what Jan Noyes, an ergonomics expert at Britain's University of Bristol, terms "a de-skilling of the crew." No one doubts that autopilot has contributed to improvements in flight safety over the years. It reduces pilot fatigue and provides advance warnings of problems, and it can keep a plane airborne should the crew become disabled. But the steady overall decline in plane crashes masks the recent arrival of "a spectacularly new type of accident," says Raja Parasuraman, a psychology professor at George Mason University and a leading authority on automation. When an autopilot system fails, too many pilots, thrust abruptly into what has become a rare role, make mistakes. Rory Kay, a veteran United captain who has served as the top safety official of the Air Line Pilots Association, put the problem bluntly in a 2011 interview with the Associated Press: "We're forgetting how to fly." The Federal Aviation Administration has become so concerned that in January it issued a "safety alert" to airlines, urging them to get their pilots to do more manual flying. An overreliance on automation, the agency warned, could put planes and passengers at risk.

The experience of airlines should give us pause. It reveals that automation, for all its benefits, can take a toll on the performance and talents of those who rely on it. The implications go well beyond safety. Because automation alters how we act, how we learn, and what we know, it has an ethical dimension. The choices we make, or fail to make, about which tasks we hand off to machines shape our lives and the place we make for ourselves in the world. That has always been true, but in recent years, as the locus of labor-saving technology has shifted from machinery to software, automation has become ever more pervasive, even as its workings have become more hidden from us. Seeking convenience, speed, and efficiency, we rush to off-load

work to computers without reflecting on what we might be sacrificing as a result.

Doctors use computers to make diagnoses and to perform surgery. Wall Street bankers use them to assemble and trade financial instruments. Architects use them to design buildings. Attorneys use them in document discovery. And it's not only professional work that's being computerized. Thanks to smartphones and other small, affordable computers, we depend on software to carry out many of our everyday routines. We launch apps to aid us in shopping, cooking, socializing, even raising our kids. We follow turn-by-turn GPS instructions. We seek advice from recommendation engines on what to watch, read, and listen to. We call on Google, or Siri, to answer our questions and solve our problems. More and more, at work and at leisure, we're living our lives inside glass cockpits.

A hundred years ago, the British mathematician and philosopher Alfred North Whitehead wrote, "Civilization advances by extending the number of important operations which we can perform without thinking about them." It's hard to imagine a more confident expression of faith in automation. Implicit in Whitehead's words is a belief in a hierarchy of human activities: Every time we off-load a job to a tool or a machine, we free ourselves to climb to a higher pursuit, one requiring greater dexterity, deeper intelligence, or a broader perspective. We may lose something with each upward step, but what we gain is, in the long run, far greater.

History provides plenty of evidence to support Whitehead. We humans have been handing off chores, both physical and mental, to tools since the invention of the lever, the wheel, and the counting bead. But Whitehead's observation should not be mistaken for a universal truth. He was writing when automation tended to be limited to distinct, well-defined, and repetitive tasks—weaving fabric with a steam loom, adding numbers with a mechanical calculator. Automation is different now. Computers can be programmed to perform complex activities in which a succession of tightly coordinated tasks is carried out through an evaluation of many variables. Many software programs take on intellectual work—observing and sensing, analyzing and judging, even making decisions—that until recently was considered the preserve of humans. That may leave the person operating the computer to play the role of a high-tech clerk—entering data, monitoring outputs, and watching for failures. Rather than opening new

frontiers of thought and action, software ends up narrowing our focus. We trade subtle, specialized talents for more routine, less distinctive ones.

Most of us want to believe that automation frees us to spend our time on higher pursuits but doesn't otherwise alter the way we behave or think. That view is a fallacy—an expression of what scholars of automation call the “substitution myth.” A labor-saving device doesn't just provide a substitute for some isolated component of a job or other activity. It alters the character of the entire task, including the roles, attitudes, and skills of the people taking part. As Parasuraman and a colleague explained in a 2010 journal article, “Automation does not simply supplant human activity but rather changes it, often in ways unintended and unanticipated by the designers of automation.”

Psychologists have found that when we work with computers, we often fall victim to two cognitive ailments—complacency and bias—that can undercut our performance and lead to mistakes. Automation complacency occurs when a computer lulls us into a false sense of security. Confident that the machine will work flawlessly and handle any problem that crops up, we allow our attention to drift. We become disengaged from our work, and our awareness of what's going on around us fades. Automation bias occurs when we place too much faith in the accuracy of the information coming through our monitors. Our trust in the software becomes so strong that we ignore or discount other information sources, including our own eyes and ears. When a computer provides incorrect or insufficient data, we remain oblivious to the error.

Examples of complacency and bias have been well documented in high-risk situations—on flight decks and battlefields, in factory control rooms—but recent studies suggest that the problems can bedevil anyone working with a computer. Many radiologists today use analytical software to highlight suspicious areas on mammograms. Usually, the highlights aid in the discovery of disease. But they can also have the opposite effect. Biased by the software's suggestions, radiologists may give cursory attention to the areas of an image that haven't been highlighted, sometimes overlooking an early-stage tumor. Most of us have experienced complacency when at a computer. In using e-mail or word-processing software, we become less proficient proofreaders when we know that a spell-checker is at work.

The way computers can weaken awareness and attentiveness points to a deeper problem. Automation turns us from actors into observers. Instead

of manipulating the yoke, we watch the screen. That shift may make our lives easier, but it can also inhibit the development of expertise. Since the late 1970s, psychologists have been documenting a phenomenon called the "generation effect." It was first observed in studies of vocabulary, which revealed that people remember words much better when they actively call them to mind—when they generate them—than when they simply read them. The effect, it has since become clear, influences learning in many different circumstances. When you engage actively in a task, you set off intricate mental processes that allow you to retain more knowledge. You learn more and remember more. When you repeat the same task over a long period, your brain constructs specialized neural circuits dedicated to the activity. It assembles a rich store of information and organizes that knowledge in a way that allows you to tap into it instantaneously. Whether it's Serena Williams on a tennis court or Magnus Carlsen at a chessboard, an expert can spot patterns, evaluate signals, and react to changing circumstances with speed and precision that can seem uncanny. What looks like instinct is hard-won skill, skill that requires exactly the kind of struggle that modern software seeks to alleviate.

In 2005, Christof van Nimwegen, a cognitive psychologist in the Netherlands, began an investigation into software's effects on the development of know-how. He recruited two sets of people to play a computer game based on a classic logic puzzle called *Missionaries and Cannibals*. To complete the puzzle, a player has to transport five missionaries and five cannibals (or, in van Nimwegen's version, five yellow balls and five blue ones) across a river, using a boat that can accommodate no more than three passengers at a time. The tricky part is that cannibals must never outnumber missionaries, either in the boat or on the riverbanks. One of van Nimwegen's groups worked on the puzzle using software that provided step-by-step guidance, highlighting which moves were permissible and which weren't. The other group used a rudimentary program that offered no assistance.

As you might expect, the people using the helpful software made quicker progress at the outset. They could simply follow the prompts rather than having to pause before each move to remember the rules and figure out how they applied to the new situation. But as the test proceeded, those using the rudimentary software gained the upper hand. They developed a clearer conceptual understanding of the task, plotted better strategies,

and made fewer mistakes. Eight months later, van Nimwegen had the same people work through the puzzle again. Those who had earlier used the rudimentary software finished the game almost twice as quickly as their counterparts. Enjoying the benefits of the generation effect, they displayed better "imprinting of knowledge."

What van Nimwegen observed in his laboratory—that when we automate an activity, we hamper our ability to translate information into knowledge—is also being documented in the real world. In many businesses, managers and other professionals have come to depend on decision-support systems to analyze information and suggest courses of action. Accountants, for example, use the systems in corporate audits. The applications speed the work, but some signs suggest that as the software becomes more capable, the accountants become less so. One recent study, conducted by Australian researchers, examined the effects of systems used by three international accounting firms. Two of the firms employed highly advanced software that, based on an accountant's answers to basic questions about a client, recommended a set of relevant business risks to be included in the client's audit file. The third firm used simpler software that required an accountant to assess a list of possible risks and manually select the pertinent ones. The researchers gave accountants from each firm a test measuring their expertise. Those from the firm with the less helpful software displayed a significantly stronger understanding of different forms of risk than did those from the other two firms.

What's most astonishing, and unsettling, about computer automation is that it's still in its early stages. Experts used to assume that there were limits to the ability of programmers to automate complicated tasks, particularly those involving sensory perception, pattern recognition, and conceptual knowledge. They pointed to the example of driving a car, which requires not only the instantaneous interpretation of a welter of visual signals but also the ability to adapt seamlessly to unanticipated situations. "Executing a left turn across oncoming traffic," two prominent economists wrote in 2004, "involves so many factors that it is hard to imagine the set of rules that can replicate a driver's behavior." Just six years later, in October 2010, Google announced that it had built a fleet of seven "self-driving cars," which had already logged more than 140,000 miles on roads in California and Nevada.

Driverless cars provide a preview of how robots will be able to navigate and perform work in the physical world, taking over activities requiring environmental awareness, coordinated motion, and fluid decision making. Equally rapid progress is being made in automating cerebral tasks. Just a few years ago, the idea of a computer competing on a game show like *Jeopardy* would have seemed laughable, but in a celebrated match in 2011, the IBM supercomputer Watson trounced *Jeopardy's* all-time champion, Ken Jennings. Watson doesn't think the way people think; it has no understanding of what it's doing or saying. Its advantage lies in the extraordinary speed of modern computer processors.

In *Race Against the Machine*, a 2011 e-book on the economic implications of computerization, the MIT researchers Erik Brynjolfsson and Andrew McAfee argue that Google's driverless car and IBM's Watson are examples of a new wave of automation that, drawing on the "exponential growth" in computer power, will change the nature of work in virtually every job and profession. Today, they write, "computers improve so quickly that their capabilities pass from the realm of science fiction into the everyday world not over the course of a human lifetime, or even within the span of a professional's career, but instead in just a few years."

Who needs humans, anyway? That question, in one rhetorical form or another, comes up frequently in discussions of automation. If computers' abilities are expanding so quickly and if people, by comparison, seem slow, clumsy, and error-prone, why not build immaculately self-contained systems that perform flawlessly without any human oversight or intervention? Why not take the human factor out of the equation? The technology theorist Kevin Kelly, commenting on the link between automation and pilot error, argued that the obvious solution is to develop an entirely autonomous autopilot: "Human pilots should not be flying planes in the long run." The Silicon Valley venture capitalist Vinod Khosla recently suggested that health care will be much improved when medical software—which he has dubbed "Doctor Algorithm"—evolves from assisting primary-care physicians in making diagnoses to replacing the doctors entirely. The cure for imperfect automation is total automation.

That idea is seductive, but no machine is infallible. Sooner or later, even the most advanced technology will break down, misfire, or, in the case of a computerized system, encounter circumstances that its designers never anticipated. As automation technologies become more complex, relying on

interdependencies among algorithms, databases, sensors, and mechanical parts, the potential sources of failure multiply. They also become harder to detect. All of the parts may work flawlessly, but a small error in system design can still cause a major accident. And even if a perfect system could be designed, it would still have to operate in an imperfect world.

In a classic 1983 article in the journal *Automatica*, Lisanne Bainbridge, an engineering psychologist at University College London, described a conundrum of computer automation. Because many system designers assume that human operators are "unreliable and inefficient," at least when compared with a computer, they strive to give the operators as small a role as possible. People end up functioning as mere monitors, passive watchers of screens. That's a job that humans, with our notoriously wandering minds, are especially bad at. Research on vigilance, dating back to studies of radar operators during World War II, shows that people have trouble maintaining their attention on a stable display of information for more than half an hour. "This means," Bainbridge observed, "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," even an experienced operator will eventually begin to act like an inexperienced one if restricted to just watching. The lack of awareness and the degradation of know-how raise the odds that when something goes wrong, the operator will react ineptly. The assumption that the human will be the weakest link in the system becomes self-fulfilling.

Psychologists have discovered some simple ways to temper automation's ill effects. You can program software to shift control back to human operators at frequent but irregular intervals; knowing that they may need to take command at any moment keeps people engaged, promoting situational awareness and learning. You can put limits on the scope of automation, making sure that people working with computers perform challenging tasks rather than merely observing. Giving people more to do helps sustain the generation effect. You can incorporate educational routines into software, requiring users to repeat difficult manual and mental tasks that encourage memory formation and skill building.

Some software writers take such suggestions to heart. In schools, the best instructional programs help students master a subject by encouraging attentiveness, demanding hard work, and reinforcing learned skills through repetition. Their design reflects the latest discoveries about how our

brains store memories and weave them into conceptual knowledge and practical know-how. But most software applications don't foster learning and engagement. In fact, they have the opposite effect. That's because taking the steps necessary to promote the development and maintenance of expertise almost always entails a sacrifice of speed and productivity. Learning requires inefficiency. Businesses, which seek to maximize productivity and profit, would rarely accept such a trade-off. Individuals, too, almost always seek efficiency and convenience. We pick the program that lightens our load, not the one that makes us work harder and longer. Abstract concerns about the fate of human talent can't compete with the allure of saving time and money.

The small island of Igloolik, off the coast of the Melville Peninsula in the Nunavut territory of northern Canada, is a bewildering place in the winter. The average temperature hovers at about 20 degrees below zero, thick sheets of sea ice cover the surrounding waters, and the sun is rarely seen. Despite the brutal conditions, Inuit hunters have for some 4,000 years ventured out from their homes on the island and traveled across miles of ice and tundra to search for game. The hunters' ability to navigate vast stretches of the barren Arctic terrain, where landmarks are few, snow formations are in constant flux, and trails disappear overnight, has amazed explorers and scientists for centuries. The Inuit's extraordinary way-finding skills are born not of technological prowess—they long eschewed maps and compasses—but of a profound understanding of winds, snowdrift patterns, animal behavior, stars, and tides.

Inuit culture is changing now. The Igloolik hunters have begun to rely on computer-generated maps to get around. Adoption of GPS technology has been particularly strong among younger Inuit, and it's not hard to understand why. The ease and convenience of automated navigation makes the traditional Inuit techniques seem archaic and cumbersome.

But as GPS devices have proliferated on Igloolik, reports of serious accidents during hunts have spread. A hunter who hasn't developed way-finding skills can easily become lost, particularly if his GPS receiver fails. The routes so meticulously plotted on satellite maps can also give hunters tunnel vision, leading them onto thin ice or into other hazards a skilled navigator would avoid. The anthropologist Claudio Aporta, of Carleton University in Ottawa, has been studying Inuit hunters for more than 15 years. He notes that while satellite navigation offers practical advantages,

its adoption has already brought a deterioration in way-finding abilities and, more generally, a weakened feel for the land. An Inuit on a GPS-equipped snowmobile is not so different from a suburban commuter in a GPS-equipped SUV: as he devotes his attention to the instructions coming from the computer, he loses sight of his surroundings. He travels "blindfolded," as Aporta puts it. A unique talent that has distinguished a people for centuries may evaporate in a generation.

Whether it's a pilot on a flight deck, a doctor in an examination room, or an Inuit hunter on an ice floe, knowing demands doing. One of the most remarkable things about us is also one of the easiest to overlook: each time we collide with the real, we deepen our understanding of the world and become more fully a part of it. While we're wrestling with a difficult task, we may be motivated by an anticipation of the ends of our labor, but it's the work itself—the means—that makes us who we are. Computer automation severs the ends from the means. It makes getting what we want easier, but it distances us from the work of knowing. As we transform ourselves into creatures of the screen, we face an existential question: Does our essence still lie in what we know, or are we now content to be defined by what we want? If we don't grapple with that question ourselves, our gadgets will be happy to answer it for us