

FREE SAMPLE

The Expensive Things

*What Stays Valuable
When Answers Get Cheap*

*The prologue, four minutes over the Atlantic, the
missing rung, the five questions and the keep-test*

Pranav S.

FREE SAMPLE · OCTOBER 2026

Contents of the full book

Everything in The Expensive Things, section by section. The parts marked INCLUDED are complete in this sample, and those marked EXCERPT are partly included. Every chapter also ends with the question it leaves you and notes on its sources.

A Note on Names and Sources

PROLOGUE	The Hard Parts INCLUDED	<i>Three things that didn't get cheap · Answers and what they're made of · Where the evidence comes from · What this book isn't · A note on how to read this</i>
----------	--------------------------------	--

PART ONE: THE MECHANISMS

CHAPTER 1	The Amateurs Who Beat the Grandmasters	<i>A better process · Written down · The floor rises · The Kenyan entrepreneurs</i>
CHAPTER 2	The Job Called "Computer"	<i>The West Area · The what-if game · Cheap code · One tax return a year</i>
CHAPTER 3	The Site That Taught the Machines	<i>The forum that went quiet · Bogle's Folly · The Drain · Prices that held</i>
CHAPTER 4	The Cockpit EXCERPT	<i>Four minutes over the Atlantic · The ironies · The same irony, new machines · Kept on purpose</i>
CHAPTER 5	Stop Training Radiologists	<i>Three explanations · What a signature is worth · Where the Stake gives way</i>
INTERLUDE	The Five Questions INCLUDED	

PART TWO: THE COLLISIONS

CHAPTER 6	The Missing Rung EXCERPT	<i>Why the beginner loses · Two economies · Inside the rung · When the rung comes back</i>
CHAPTER 7	The Collapse of Proof	<i>What a grade is for · The Flood at the door · What gets staked instead · Where proof survives</i>

CHAPTER 8	Homework After Answers <i>Four sessions in Turkey · Performance and learning · The tutors that worked · The keep-test for children, and where it fails</i>
CHAPTER 9	The Arbitrage <i>Eighteen seats · What crossed the wire · The first person replaced · Where the line moves</i>
CHAPTER 10	The Median Machine <i>Attention is the scarce thing · The Flood, counted · When the name is the product · The library still works</i>
CHAPTER 11	Baumol's Revenge <i>Four players, forty minutes · The price of a body · Relative prices are not wages · What would break it</i>

PART THREE: THE LIMITS

CHAPTER 12	What If It Doesn't Stop? <i>The entity · Three ways to be wrong · The best case for each side · What breaks first</i>
EPILOGUE	Back to the Hard Parts <i>The guide, through five questions · How this book was made · The ironies, owned · What the evidence says so far · What's left</i>
FIELD GUIDE	Running the Five Questions EXCERPT <i>Before you start · The five questions · The keep-test · For three kinds of reader · Three worked examples</i>

Also by Pranav S.

About this edition

PROLOGUE

The Hard Parts

The first version of an earlier book of mine took almost no time to make and was worth nothing.

It was a long PDF about using an AI coding tool, and it had everything a guide is supposed to have: chapters, headings, numbered tips, a confident tone. It also had almost nothing in it. The sentences were fluent and the advice was generic, the kind that would apply to any tool and help with none. When I went back through it, one idea in the whole document was worth keeping. Everything else was padding that looked like expertise. Of its 1,668 sentences, only 183 were different, and one of them appeared 128 times.

The second version became a real book: 19 chapters, an afterword and six appendices, close to 100,000 words, more than 400 pages as a PDF. I wrote it with heavy help from AI tools, and I built the website that sells it the same way. A great deal of what would once have taken a small team, or one person a very long time, arrived quickly: chapter drafts, code samples, the pages of the site, the checkout, the emails. The first full draft of the book and the site that sells it took about a day. Checking and revising them took longer.

That's the part of the story people expect, and it's true. This prologue is about the other part. The pieces that mattered most didn't get faster. Some got slower. And when I looked closely at which pieces those were, I found the argument of this book.

Three things that didn't get cheap

The first hard part was deciding what to build. The tool would happily produce a chapter on any topic I named, at any length, in any tone. What it couldn't tell me was which topics a working developer would actually pay to read about, which ones they already knew, and which ones the official documentation already covered better than I could. What separated the worthless PDF from the real book was a long series of choices about what to put in, what to cut and what order a reader needed things in. The sentences were never the hard part. Those choices came from somewhere other than the tool: from knowing the reader, from having been stuck in the same places, from reading the drafts as a skeptic would.

That kind of skill is hard to write down. It's learned by doing the work.

The second hard part was getting a stranger to trust the thing. A book from an unknown author, sold on its own website, asks a reader to take a risk. Nothing about the text could remove that risk, because a reader can't judge the whole text until they've paid for it. So every decision about trust had to be made by me, and none of them could be generated. I had to decide what name would go on the cover and whether it would be my own, what to charge, how much of the book to give away free so a reader could judge it before buying, whether to offer a refund and for how long, whether to promise free updates when the tool changed, and who would answer when someone wrote in to say something was wrong.

Each of those questions is a small promise. Each one puts something of mine on the line: my name, my time or my money. Some of them I still hadn't settled when this book began. In the end I chose a free sample of about 25 pages, a 30-day refund and corrections reported through the site that sells the book. They were the slowest decisions in the whole project, and no amount of cheap drafting could make them for me.

The third hard part was knowing whether any of it was right. A technical guide is useless if its commands are wrong, and AI tools are fluent in exactly the way that makes a wrong command look right. So every instruction, setting and claim

in the book had to be checked against the tool's actual documentation, and the checking took longer than the writing. The guide was also built on top of other people's work: the documentation its makers wrote, and the experience of developers who had written publicly about what worked. My book drew on all of that. It's fair to ask what happens to those sources if readers stop reading documentation and forums because guides like mine, or chatbots like the one I used, give them the answer directly.

Choosing what's worth making, staking something so that strangers will trust it, and checking that it's true: those were the hard parts. They were also, I came to think, the expensive things. The words had become cheap. What made the words worth reading had not.

Answers and what they're made of

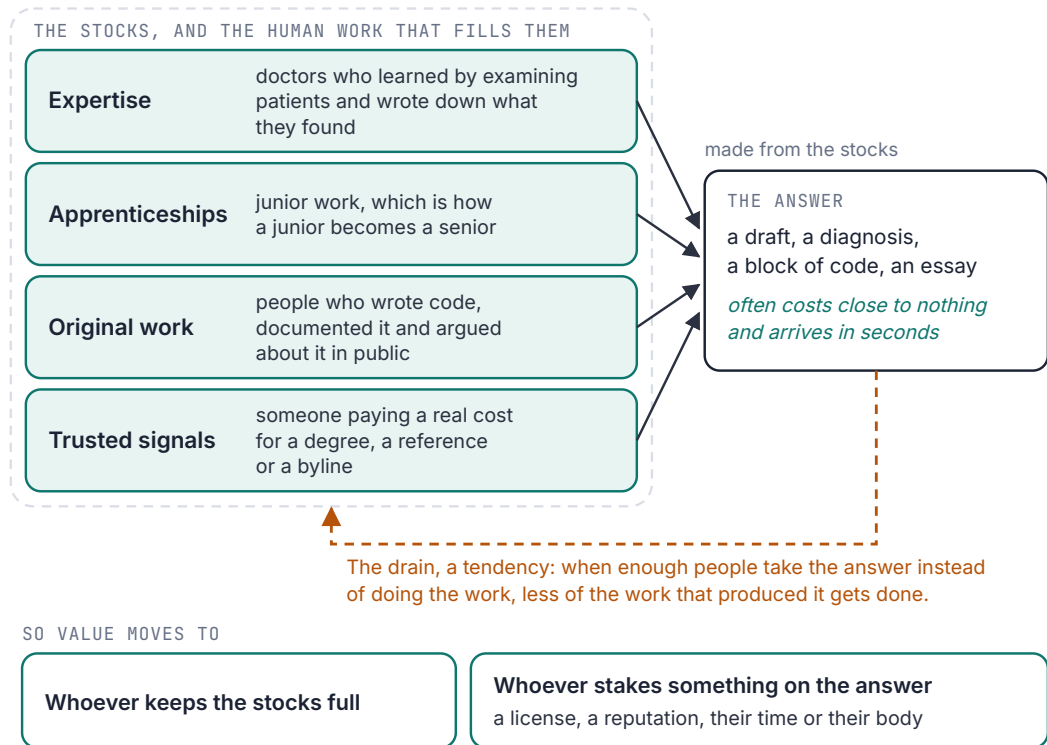
That distinction runs through the whole of this book.

An answer, in the sense I'll use the word, is any output: a draft, a diagnosis, a block of code, an essay, a legal summary, a translation, a design. Answers are getting cheap very quickly. For many kinds of work, a competent first answer now costs close to nothing and arrives in seconds.

But answers are made from something. A good answer about code is distilled from the work of people who wrote code, documented it and argued about it in public. A good diagnosis rests on generations of doctors who learned by examining patients and wrote down what they found. A trustworthy signal, such as a degree, a reference or a byline, is trustworthy because someone, somewhere, paid a real cost to produce it. I'll call these things stocks, because they behave like reserves built up over time: stocks of expertise, of apprenticeships, of original work and of trusted signals.

This book's argument is that cheap answers draw those stocks down without refilling them. When enough people take the answer instead of doing the work, less of the work that produced the answer gets done. When nobody does junior

work anymore, nobody becomes a senior. When polished output is free, polish stops proving anything about the person who sent it. So value moves to whoever keeps the stocks full, and to whoever stakes something on the answer: a license, a reputation, their time or their body.



Where an answer comes from: four stocks, each kept full by human work, and the answer made from them. The dashed drain marks a tendency, and nothing in the figure is sized.

That claim could be wrong, and the book takes seriously the ways it might be. Machines might learn to refill the stocks themselves, as they already have in games like chess. Markets might reward the people who refill them quickly enough that the drain corrects itself. Cheap answers might even create more original work than they consume. Chapter 12 tests each of those possibilities. But the evidence so far, from a surprising range of places, points in one direction often enough to be worth following. Some of the best-measured cases point the other way, and the book reports those too.

Where the evidence comes from

This book mostly looks sideways and back, at the fields where machines got good first.

Chess players have shared the game with machines at least since 1997, when IBM's Deep Blue became the first computer to beat a reigning world champion, Garry Kasparov, in a match under standard tournament controls. Airline pilots have flown alongside automation for longer than that. Accountants absorbed the spreadsheet, investors absorbed the index fund, and radiologists absorbed a confident prediction that they were about to be replaced. Each of these is a small world in which something like cheap intelligence arrived early, and each of them has a record of what happened next. Some are direct cases. Others are neighboring technologies or analogies, and I'll say which is which as we go.

From those cases come five patterns, each of which answers a question you can ask about your own work:

- **Who gains?** The Rising Floor: AI lifts novices toward experts, and what stays scarce is the judgment AI can't supply.
- **Does the work grow or shrink?** The Flood: when something gets cheap and demand can grow, far more of it gets made.
- **What gets drained?** The Drain: when enough people take the shared answer, fewer do the work it was built from.
- **What will I lose?** The Forgetting: automation takes the routine cases and leaves you the hard ones, just as your skills fade.
- **What still needs a person?** The Stake: something proves its value only when someone puts something on the line.

Part One introduces the patterns one at a time, each through the field where it showed up first. Part Two watches them collide in the places most readers live: careers, hiring and school, the outsourcing industry that serves much of the

world, culture and the wider economy. Part Three asks what happens if the machines don't stop improving, and whether the patterns survive. At the end, the five questions come back as a diagnostic you can run on your own work.



Map of the book. Part One finds each pattern in the field where it showed first, Part Two watches the patterns collide two at a time, and Part Three tests whether they survive machines that keep improving.

What this book isn't

It isn't a forecast. I don't know what AI will be able to do in five years, and I'm suspicious of anyone who says they do. The patterns in this book are useful precisely because they don't depend on any particular model or product. They de-

scribe what tends to happen when answers get cheap, whatever is producing them.

It isn't a guide to using AI tools. This book is about the consequences that reach you whether or not you ever open a chatbot.

And it isn't a lament. Cheap answers are an extraordinary thing. They put a competent first draft, a second opinion and a patient tutor within reach of people who could never have afforded them. Some of the stories in this book are hopeful, and the hopeful ones are as real as the others. The point isn't that cheap answers are bad. It's that they have costs that are easy to miss, and that the people who see those costs early will make better decisions about their work, their children's education and their own skills.

A note on how to read this

Each chapter opens on a puzzle, a fact that shouldn't be true. Then it tells the case behind the puzzle, explains the mechanism and credits whoever first described it, looks at today's evidence and asks where the pattern breaks. Each one ends with a question addressed to you.

The stories come from published sources, and the notes at the end of each chapter point to every one of them. The note at the front of the book explains who is named and who isn't.

You may be wondering whether AI wrote this book. It's a fair question for a book on this subject, and I'll answer it at the end, using the five questions, because the answer turns out to be part of the argument.

Start where the machines started: with a game.

Notes

1. Deep Blue's defeat of Garry Kasparov in May 1997, winning the six-game rematch 3.5–2.5, the first time a computer beat a reigning world champion in a match under standard tournament controls. IBM, "Deep Blue," IBM History. <https://www.ibm.com/history/deep-blue>

The Cockpit

Commercial flying keeps getting safer. In the decade from 1968 to 1977, the risk of dying on a commercial flight worked out to about one death for every 350,000 passenger boardings, according to an analysis by MIT's Arnold Barnett, a professor of management science, and Jan Reig Torra. From 2018 to 2022, it was about one for every 13.7 million. The risk has fallen by roughly half every decade.

Automation is part of that story, though not all of it. Barnett's analysis measures the outcome without handing out credit, and MIT's summary names collision-avoidance systems, training and the work of regulators among the reasons. Regulators treat it as a safety tool too: a Transport Canada advisory on the subject says automatic flight-control systems give excellent results in most situations.

That's what makes one night in 2009 so hard to explain. On 1 June, Air France flight 447, an Airbus A330 flying from Rio de Janeiro to Paris with 228 people aboard, went down in the Atlantic. The aircraft hit the sea 4 minutes and 23 seconds after the autopilot disconnected, and everyone on board died. According to the French investigators, the engines worked normally throughout, the control surfaces moved as the pilots commanded, and the airspeed readings that had failed came back within about a minute.

If automation made flying so much safer, how did a trained crew lose an aircraft that was still answering to them?

Four minutes over the Atlantic

Air France 447 comes from a neighboring technology. The autopilot is the longest-running experiment we have in handing skilled work to a machine while keeping a person on hand for the moment the machine lets go. Everything below comes from the final report of the BEA, France's civil aviation safety investigator, published in July 2012, and the account stays with what the report records.

The flight took off from Rio on the evening of 31 May. Because the trip was long, the crew had been augmented with a second copilot so the pilots could rest in turns. Shortly after 02:00 UTC, the captain left the cockpit for his rest. The BEA notes that he left no specific instructions for crossing the storm zone ahead, and that his relief was designated only implicitly. The copilot left in the right seat to fly the aircraft had 2,936 flying hours. The copilot beside him, in the left seat, had 6,547.

The airline's operations center had told the crew to expect storm activity on their route, tied to the intertropical convergence zone. The weather was not exceptional for June in that region, the BEA found, but there were powerful clusters of storm cloud on the route, and several aircraft flying near the same altitude changed course to avoid them. AF447's crew turned 12 degrees to the left.

At 02:10:05, the speed readings went wrong. The BEA concludes that ice crystals most likely obstructed the aircraft's Pitot probes, which measure one of the two air pressures from which the aircraft calculates its speed. The autopilot disconnected, then the automatic thrust. The flight controls switched from what the report calls normal law to alternate law, a mode with fewer of the built-in protections that normally keep the aircraft within its limits. The right-seat copilot, the pilot flying, announced that he had the controls.

The flight recorders preserved what came next, and the BEA sets it out in order. As the aircraft rolled right, the pilot flying answered with quick, large movements of the sidestick, swinging it nearly from one limit to the other, and pulled the nose up until the pitch reached 11 degrees in ten seconds. The other copilot

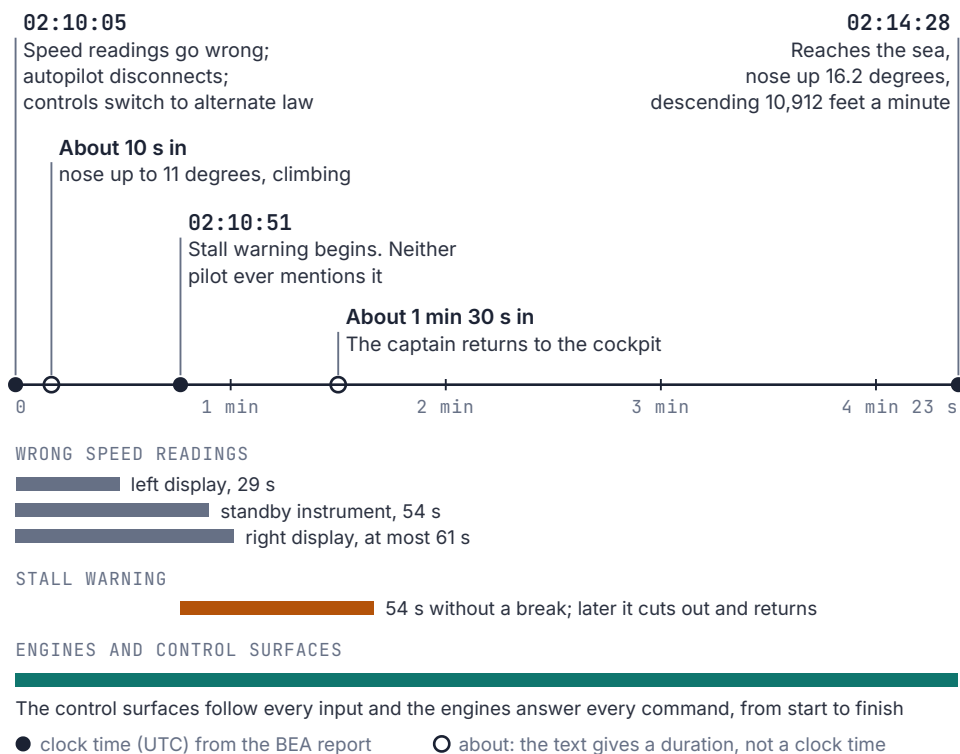
called out that the speeds were lost, read the warning messages aloud in a manner the report calls disorganized, and told the pilot flying several times that they were climbing and should come down. The aircraft rose to about 38,000 feet, losing speed as it climbed.

Within a minute of the disconnection, the aircraft had left its flight envelope, the band of speeds within which it could keep flying at that height, whose lower edge is the stall. The stall warning began at 02:10:51 and sounded without a break for 54 seconds. Soon after it started, the pilot flying pushed the thrust levers to takeoff power and pulled the nose up again. The BEA found that neither pilot ever mentioned the warning, or the shaking of the airframe that came with it, and neither recognized that the aircraft had stalled.

The captain returned to the cockpit a minute and a half after the disconnection. In the seconds that followed, the angle of attack, the measure the stall warning relies on, passed 40 degrees, and the aircraft was descending at about 10,000 feet a minute. It reached the sea at 02:14:28, nose up at 16.2 degrees, descending at 10,912 feet a minute. The crew sent no emergency message.

The wreckage lay 3,900 meters down. Finding it took nearly two years, and the flight recorders came up 23 months after the crash.

The puzzle turns on what the BEA found about the aircraft itself. From start to finish, its control surfaces followed the pilots' inputs and its engines answered every command. The wrong speeds lasted 29 seconds on the left-hand display, 54 on the standby instrument and at most 61 on the right. The airspeed fault was brief. What it set off was not.



Air France 447, from the autopilot's disconnection to the sea, as the BEA's final report records it. Times marked "about" are given in the text only as time after the disconnection, and the report's several causes, from iced speed sensors to training, are not reduced here to one.

That doesn't make the accident a story about bad pilots, and the BEA is careful not to tell it that way. Its causes run from the aircraft's sensors all the way to the training system.

The report's list is long. Pitot icing in cruise was known but poorly understood, and Airbus and Air France had already chosen a replacement probe after earlier airspeed losses; the first aircraft got one two days before the crash. In the cockpit, no message plainly said the speeds disagreed, and the angle of attack wasn't displayed. By design, the stall warning ignored angle-of-attack readings once measured speed fell below 60 knots, so when the aircraft was deep enough in the stall for that to happen, the warning went quiet, and it sounded again when the pilot flying lowered the nose. The BEA also cites flight director cues,

the steering commands on each pilot's display, that may have told the crew they were doing the right thing when they weren't, and a poorly handled startle that put both copilots under heavy emotional strain. Their cooperation, it says, was willing but chaotic.

Then there is training. Neither copilot had practiced, in flight at high altitude, either the procedure for unreliable airspeed or flying by hand, the BEA found. Their hands-on stall practice dated from basic training and from one or two simulator sessions during the training that first qualified them on the Airbus A320, flown at low altitude, with the warning expected and the response rehearsed. Current training, the report says, didn't make up for the lack of any hand-flying at high altitude, and it made basic airmanship harder for pilots to build or keep. Its recommendations put the lesson bluntly: manual handling, the BEA wrote, "cannot be improvised."

The BEA's deepest point is about design. The safety model assumed that if airspeed information failed, the crew would first keep the aircraft under control using basic airmanship, then quickly diagnose the problem and find the right procedure. On AF447, the first step failed, so the second never came. The report calls this a common failure mode in the safety model itself: losing control of the flight path left the crew unable to work out what was happening, and so unable to reach the procedure meant to fix it.

So the BEA's account has several strands: ice, a probe the industry was already replacing, a warning that fell silent at the worst moment, flight director cues that may have misled, a crew whose coordination broke down and a training gap. Its 41 safety recommendations ranged as widely, and one asked the European Aviation Safety Agency to make regular hand-flying practice in approaching and recovering from a stall mandatory, including at high altitude.

The BEA states that its investigations exist to improve safety and aren't meant to assign blame or liability. That question went to the French courts. In April 2023, a Paris criminal court acquitted Air France and Airbus. On 21 May 2026, the Paris Court of Appeal overturned the acquittal, found both companies guilty

of corporate manslaughter and fined each €225,000, the maximum. Both have denied the charges, and both announced appeals to France's highest court. This chapter draws only on the BEA's safety findings.

Of the BEA's strands, this chapter follows one, lost manual skill, because it's the strand that automation produces simply by working well.

The ironies

The pattern under AF447 was described a quarter century earlier, by someone writing mainly about other kinds of machines.

In 1983, Lisanne Bainbridge, a psychologist at University College London, published a short paper called "Ironies of Automation" in the journal *Automatica*. She was writing chiefly about industrial control rooms, where operators supervise automated process plants, though she drew some examples from flight decks.

One of her ironies concerns the designer. A designer who sees the human operator as unreliable tries to automate the operator out of the system, and still has to leave the operator the jobs nobody could work out how to automate. The person in the control room ends up with an arbitrary collection of leftovers, which come in two kinds: watching the automation, and taking over when it fails.

Taking over is where another irony bites. Stepping in requires manual skill, and as Bainbridge put it, "physical skills deteriorate when they are not used."

Someone who was once an expert operator, but has since only watched the automation run, may now be a novice. And because takeover happens when something has already gone wrong, the person stepping in needs more skill than usual, and a lighter load, at exactly the moment when neither is likely.

Watching is no easier. Bainbridge cites vigilance research: even a highly motivated person can keep effective watch over an uneventful display for only about half an hour. And there's a further twist. The automatic system was put in be-

cause it outperforms the operator, yet the operator is asked to confirm that it's working properly.

Then she makes the point that connects most directly to this book. There was some concern, she wrote, that the automated systems of her day were being watched by people who had once run the plants by hand, and depended on skills that the next generation couldn't be expected to have. If so, the skill was a stock, built up in an earlier era of hands-on work, and automation was spending it without refilling it.

The irony she tentatively called the final one is the one aviation would rediscover. The best automated systems, the ones that rarely need a person to step in, may be the ones whose operators need the most training. When failures are rare, the failures are the only practice anyone gets.

This is the fourth of the book's patterns, the Forgetting, and its question is: what will I lose? Automation takes the routine cases. Skills fade without practice. People are left with the rare, hard cases they've had the least practice for.

Credit for bringing this argument to general readers belongs to Nicholas Carr. His book *The Glass Cage* (2014) made the case years before ChatGPT. It opens with the same FAA alert this chapter comes to below, and its examples include deadly crashes blamed on pilots leaning too heavily on the autopilot. This chapter walks a path Carr cleared, and adds what has been measured since.

Aviation had been naming the problem for years before that. In a 1997 training presentation at American Airlines, Captain Warren Vanderburgh coined the phrase "children of the magenta," after the magenta course line that flight computers draw on a pilot's display. It became shorthand for pilots who are good at running the systems but short on the stick-and-rudder skill to fly the airplane themselves. His advice, as the training magazine of the Aircraft Owners and Pilots Association summarizes it, was to use the level of automation the moment calls for, and if the automation stops making sense close to the ground, to disconnect it and fly.

In January 2013, the US Federal Aviation Administration issued a Safety Alert for Operators, SAFO 13002, titled “Manual Flight Operations.” It warned that continuous use of autopilots and autothrottles could weaken a pilot’s ability to recover quickly from an undesired aircraft state, cited operational data showing an increase in manual handling errors, and encouraged airlines to build manual flying into both line operations and training when appropriate. Aviation’s answer to the Forgetting, in other words, was to schedule practice on purpose.

NOTE: The full chapter goes on to the same irony in today’s AI tools and the skills worth keeping on purpose, and ends with the chapter’s question and its source notes.

INTERLUDE

The Five Questions

Part One followed five patterns into the fields where each first showed itself. Before Part Two puts them to work, here they are side by side. Together they make the diagnostic the rest of the book applies, and the one you can carry into your own work.

1. Who gains? *The Rising Floor* (Chapter 1). AI supplies codified skill, the kind that can be written down, so novices' work climbs toward experts'. What stays scarce is tacit skill: choosing the problem and judging the answer. It fails when novices can't tell good output from bad, because then AI widens the gap instead of closing it.

2. Does the work grow or shrink? *The Flood* (Chapter 2). When making something gets cheaper and demand can grow, far more of it gets made. The field can grow even while particular jobs vanish. It fails when demand is fixed: nobody wants more of something just because it got cheaper to produce.

3. What gets drained? *The Drain* (Chapter 3). A shared answer is built from original work. When enough people take the answer instead of doing the work, the work slows, and the answer holds for a while before it weakens. Outputs converging is the early symptom. It fails when the shared answer is better than the best humans, when the stock refills itself, or when scarcity pays people to refill it.

4. What will I lose? *The Forgetting* (Chapter 4). Automation takes the routine cases. Skills fade without practice, and people are left with the rare, hard cases they're least practiced for. It fails when the system never breaks in ways that

matter, or when the skill is cheap to relearn. Chapter 4’s keep-test sorts any skill into three piles: hand it off, keep it as a backup, or keep it at the core.

5. What still needs a person? *The Stake* (Chapter 5). Something proves its value only when someone puts something on the line: a license, a reputation, their time or their body. Cheap answers stake nothing, so value moves to whoever stakes something on them. It fails when errors cost too little to matter, when the law makes the tool’s maker answerable, or when a stake can be faked. And it protects the role, not the headcount: one signer can sign for far more output.

	Rising Floor	Flood	Drain	Forgetting	Stake
Rising Floor <i>Who gains?</i>	Ch 1 chess		Ch 6 careers	Ch 8 school	Ch 9 outsourcing
Flood <i>Does the work grow or shrink?</i>		Ch 2 computing	Ch 10 culture		Ch 7 hiring Ch 11 economy
Drain <i>What gets drained?</i>			Ch 3 programmers' forum		
Forgetting <i>What will I lose?</i>				Ch 4 the cockpit	
Stake <i>What still needs a person?</i>					Ch 5 radiology

☒ Part One: one pattern, in the field where it showed first
 ☐ Part Two: two patterns at once, in a world most readers live in

Five patterns, five pairs. Part One takes the patterns one at a time, on the diagonal; each Part Two chapter puts two together, and Chapters 7 and 11 share a pair, the Flood and the Stake, which Chapter 7 follows into proof and Chapter 11 into prices.

Part One borrowed each of these mechanisms from whoever found it. Part Two is where they collide, two at a time, in the worlds most readers live in. The collisions are where this book adds most of its own argument, and they’re where the stakes for you are highest.

The Missing Rung

In August 2026, the Stanford economists Erik Brynjolfsson, Bharat Chandar and Ruyu Chen released a revision, dated 12 August, of a working paper that has followed AI's effect on jobs for a year. Using records from the payroll company ADP covering millions of American workers through June 2026, they found that employment of 22-to-25-year-olds in the occupations most exposed to AI stood 19% below where it would have been had it kept pace with their peers in less-exposed jobs. Experienced workers in the same occupations showed no comparable gap.

That shouldn't be happening. Chapter 1's evidence said that AI does the most for the people who know the least, lifting a novice's output toward an expert's. If any group stood to gain from cheap answers, it was the beginner, and beginners are the ones not being hired. Both findings are well supported, and they don't contradict each other. They may be two halves of one mechanism.

This is the first of Part Two's collisions, in which two of the book's patterns act at once, and it runs as follows. The Rising Floor (who gains?) says AI lifts novices because it supplies codified skill, the kind that's written down in manuals and examples. The Drain (what gets drained?) says a shared stock runs down when everyone draws on it and nobody refills it.

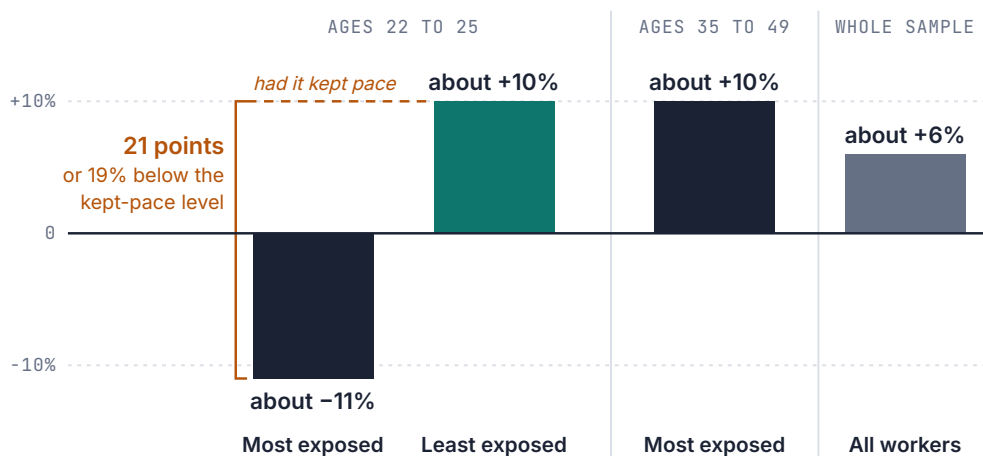
Put them together and the puzzle resolves into something more uncomfortable than a paradox. The codified skill AI hands to novices is the very thing that used to make novices worth hiring. And the stock that drains is the supply of future experts.

The paper behind the 19% calls young workers possible canaries in the coal mine, the first to show a danger that may later reach others. Where Chapter 5's radiologists were the canary that lived, these may be the ones that don't. It's a direct case, but a descriptive one, and its authors say so plainly: the patterns are early indicators, not causal estimates. The paper shows who is and isn't being employed. It doesn't prove that AI is the reason.

Its first finding is the one most often skipped. The authors find no sign that AI is displacing jobs across the economy as a whole. Average employment in their sample rose about 6% between November 2022, when ChatGPT arrived, and June 2026.

The damage is concentrated. In the two most AI-exposed of five groups of occupations, employment of 22-to-25-year-olds fell about 11% over that period. In the other three groups, their employment grew about 10%. That divergence of 21 percentage points is where the 19% comes from: measured against the higher level their employment would have reached had it kept pace, the shortfall is 19%. Meanwhile, workers aged 35 to 49 in the most exposed occupations grew in number by about 10%. The exposed occupations include software development and customer support, but the pattern persists when technology firms and computer jobs are left out, and a similar divergence shows up in jobs that don't require a college degree.

CHANGE IN EMPLOYMENT, NOVEMBER 2022 TO JUNE 2026



Most exposed: the two most AI-exposed of five groups of occupations. Least exposed: the other three.

Change in employment in ADP payroll data from November 2022 to June 2026, from the Stanford working paper. The 21-point gap is a 19% shortfall against the level young workers' employment would have reached had it kept pace with less-exposed peers, not a fall of 19%, and the authors call the patterns descriptive, early indicators rather than causal estimates.

The gap opened mainly through hiring. Young workers already in these jobs weren't being pushed out faster than before. Fewer were being let in, which the authors describe as a narrowing entry point rather than the displacement of people already employed. Since 2022, hiring and separation rates have both fallen across the sample, a low-hire, low-fire market that may be hardest on people trying to get in. Among young workers, exits from the most exposed occupations dropped by at least as much as exits from the least exposed ones. Mass displacement would have pushed them the other way. The gap is in who gets hired.

Each revision of the paper has found a wider gap, though the versions measure it in slightly different ways. The caveats are the authors' own. The patterns weaken when the analysis controls for education. Some of the divergence began before generative AI, and the gaps are larger in the ADP sample than in national survey data. Against that, the authors note that the gap kept widening long after

interest rates peaked, and that it survives controls for remote work. The fair reading is the one they give: the evidence is consistent with AI having begun to affect entry-level employment, and it isn't yet proof.

Why the beginner loses

The authors offer an explanation, drawing on earlier work by the economists Daron Acemoglu and David Autor, and they're careful to call it only one possibility. AI may be better at replacing what can be learned from books and written procedures, the material of schooling, than the know-how that comes from years on the job. If so, the tool takes over the routine, checkable work that juniors were once hired to do, and makes seniors more productive at the same time.

For anyone who writes code, the timing is hard to miss. AI's scores on software-engineering benchmarks jumped from 4.4% to 71.7% between 2023 and 2024, according to figures the paper cites. A benchmark measures the part of programming that can be specified and checked. On the authors' explanation, that is much the same part that used to justify hiring new graduates.

Two of the paper's tests bear on the explanation. The first tells automation from assistance, using data on how people actually use one AI assistant, published by its maker as the Anthropic Economic Index. The declines are concentrated where people use AI to automate tasks. Where they use it to complement their own work, jobs held steady or grew, most of all for experienced workers. And the adjustment shows up in the number of jobs, not in base pay.

The second test scored 7,514 job titles for how much each relies on codified and tacit knowledge, using an AI model to rate the job descriptions. Slower growth for young workers went with more codified knowledge, and faster growth for experienced workers went with more tacit knowledge. The result is suggestive at most. The codified-knowledge link fades once the authors control for education, and a measure built by a language model is another reason to hold it loosely.

This is where the Rising Floor turns over. In Chapter 1, the floor was good news. AI supplies codified skill, so a novice's output climbs toward an expert's, and what stays scarce is the tacit part: knowing which problem matters and whether an answer is right.

The same fact reads differently from the employer's side of the desk. Juniors arrive with what formal education gives them, which is mostly codified knowledge: the standard methods and the textbook answers. They were hired, in large part, for that, and because they were cheap. Once a tool supplies codified skill to everyone, the junior's main asset is no longer scarce, and the asset that is scarce, judgment, is the one a junior doesn't have yet.

Chapter 1's failure case sharpens the point. Among 640 Kenyan entrepreneurs given AI advice, there was no detectable average effect; the low performers did nearly 10% worse, while the high performers may have gained more than 15%. The floor rises only for people who can tell good output from bad. That cuts against the hope that a firm can swap a trained junior for an untrained one with a chatbot. Someone in the room still has to know when the tool is wrong.

In this book's vocabulary, tacit skill is judgment learned by doing. Mostly, it's learned by doing the codified work, badly at first, under someone who can see what's wrong with it. The junior years were never only a source of cheap output. They were where the next generation of seniors got made. Take away the bottom rung of a ladder and the ladder still stands, but fewer people start climbing it.

In 1964, the economist Gary Becker explained why this arrangement was always fragile. In *Human Capital*, he set out what happens to general training, meaning skills that make a worker more valuable to many employers beyond the one doing the training. Firms won't pay for general training, Becker argued, because a trained worker's wage follows their skills to any employer. A firm that trains you and keeps your pay low will lose you to a rival offering the market rate. A firm that raises your pay to match your new skills has paid for the training and gets nothing back. So trainees pay for general training themselves, through lower wages while they learn.

Most junior jobs have run on a version of that bargain. The junior earns less than a fully trained worker and, in exchange, gets supervised practice. Two economists, Luis Garicano of the London School of Economics and Luis Rayo of Northwestern's Kellogg School of Management, have recently modeled what AI does to it. Their 2025 working paper starts from the observation that apprenticeships let juniors pay for training by doing menial work, and that AI now does an increasing share of that work.

Rayo put it more vividly in an article about the research published by his school in April 2026: “the currency that apprentices are using to buy knowledge is vanishing.” In that article he describes the master's side of the old deal: the master trains the apprentice and, in return, pays them less than they produce for a while. Their model assumes that apprentices can't pay for training up front or credibly promise to pay later, so the scut work is the payment.

In their model the outcome is a race. AI removes the routine work that paid for training, which pushes firms away from apprenticeships. It also raises the ceiling on what a trained person can do, which makes training more valuable. In their law-firm example, AI lets junior associates take on advanced work sooner, so their training finishes faster and the firm has less time to recover its cost.

In their model, an apprenticeship is guaranteed to stay at least as profitable as it was before AI when a trained expert working with AI is worth more than about 2.7 times what the AI produces alone. The threshold, as it happens, is Euler's number. Below it, training gets shorter as the expert has less to pass on, and further advances in AI threaten to collapse apprenticeship altogether. It's a working paper, and the threshold is a property of a model; nobody has measured it. Its general lesson holds, though: training is safe while the gap between the trained person and the tool stays wide, and at risk as it narrows.

NOTE: The full chapter goes on to the two economies of training, what happens inside the rung and when it comes back, and ends with the chapter's question and its source notes.

Running the Five Questions

The keep-test

The keep-test, introduced in Chapter 4, turns the Forgetting into a way of sorting your own skills. Any skill can go to one of three places.

Hand it off. This fits when the tool doesn't fail in ways that matter, or when the skill is cheap to relearn if it does.

Keep it as a backup. The tool usually does it well, but when it fails, the failure matters and you'd need the skill fast. A backup skill has to be practiced on purpose now and then, the way pilots are urged to keep flying by hand (see Chapter 4), because a skill used only in emergencies is weakest exactly when the emergency comes.

Keep it at the core. This fits when the skill is the thing being paid for, or when it's what lets you judge the tool's output at all.

A short way to sort: write down eight to ten things you actually did at work last week, and ask four questions of each.

- If the tool did this badly, would I notice?
- If the tool were unavailable tomorrow, what would that cost, and whom?
- How long would it take me to get this skill back if I lost it?
- Is my name, license, reputation or presence attached to the result?

Then read the answers together. If a bad job would matter and you wouldn't notice it, the task probably isn't ready to hand off, and the skill that would notice may belong at the core. If a failure would be costly and the skill slow to recover,

it probably belongs at least in the backup pile. If the skill is what's being paid for (your name is on the result, say) or it's how you judge the tool's output, it likely belongs at the core. What's left is a candidate for handing off.

THREE PLACES A SKILL CAN GO

Hand it off

The tool doesn't fail in ways that matter, or the skill is cheap to relearn if it does.

Keep it as a backup

The tool usually does it well, but a failure would matter and you'd need the skill fast. Practice it on purpose.

Keep it at the core

The skill is what's being paid for, or it's what lets you judge the tool's output at all.

FOUR QUESTIONS FOR EACH THING YOU DID AT WORK LAST WEEK

- If the tool did this badly, would I notice?
- If the tool were unavailable tomorrow, what would that cost, and whom?
- How long would it take me to get this skill back if I lost it?
- Is my name, license, reputation or presence attached to the result?

The keep-test: three places a skill can go, the condition for each, and the four questions that sort a week's tasks among them.

The sort isn't permanent, and it is worth redoing whenever your tools or your work change. Chapter 12 describes the signs that would shift it most.

NOTE: The full Field Guide also walks through the five questions step by step, gives advice for three kinds of reader, and works through three examples.

THE FULL BOOK

Get the Full Book

This sample is a small slice of the book. The full edition has 12 chapters in three parts, an interlude, an epilogue and a field guide, from chess and the cockpit to hiring, school, culture and prices.

What you get

- The complete book as a PDF, with clickable bookmarks, and as an EPUB for e-readers
- All 12 chapters in three parts: the mechanisms, the collisions and the limits, with an interlude and an epilogue
- The Field Guide: the five questions, the keep-test, advice for three kinds of reader and three worked examples
- Notes on the sources at the end of every chapter

How to buy

1. Go to aiagentsforhumans.com/the-expensive-things and choose Get the book.
2. Pay through the secure checkout, run by Dodo Payments, which handles payment and any sales tax, VAT or GST.
3. Your download link arrives by email straight away, and you can download the book again any time from the Dodo Payments customer portal.

The Expensive Things: Free Sample. A free sample of *The Expensive Things: What Stays Valuable When Answers Get Cheap*, first edition, October 2026. © 2026 Pranav S. You're welcome to share this sample with anyone, as long as it stays unchanged and free.