

THE RESEARCH LEDGER

The AI-Native Professional

An Introduction to Judgment, Workflow,
and Institutions in the Future of Work

First Edition

Contents

Preface	xv
How to Use This Book	xvii
1 The Paradigm Shift	1
1.1 From instrument to interlocutor: three eras of computational work	2
1.2 What makes a general-purpose model different	4
1.3 Models, systems, and agents	4
1.4 The economics of a new substrate	6
1.5 Augmentation, automation, and the verifiability frontier . .	9
1.6 What “AI-native” means (and does not mean)	10
1.7 The argument of the book	11
2 The Anatomy of an AI-Native Professional	15
2.1 Why skill lists fail: from tool proficiency to disciplined practice	16
2.2 Capability I: problem framing	17
2.3 Capability II: delegation design	18
2.4 Capability III: verification	19
2.5 Capability IV: context engineering	20
2.6 Capability V: ethical stewardship	21
2.7 Calibration: appropriate reliance	21
2.8 Cognitive costs: deskilling, offloading, and metacognitive laziness	24
2.9 A worked anatomy: one task through the five capabilities . .	24
2.10 Self-assessment and the honesty problem	27
3 Workflows: From Single-Step to Multi-Step Delegation	31
3.1 The unit of redesign is the task, not the job	31
3.2 Task decomposition and the delegation ladder	32
3.3 From prompt engineering to context engineering	35
3.4 Retrieval, memory, and grounding	36
3.5 Verification architecture	36
3.6 Checkpoint placement	39
3.7 Human–agent teaming patterns	39
3.8 Measuring whether it worked: metric design and Goodhart traps	41
3.9 Anti-patterns	41

3.10	Measuring whether it worked: the checklist	42
4	Domain-Specific AI Transformations	45
4.1	A comparative method	46
4.2	Software engineering: code generation and the review obligation	47
4.3	Medicine: diagnosis, documentation, and the standard of care	48
4.4	Law: verification and the duty of competence	49
4.5	Scientific research: the integrity frontier	50
4.6	Finance: model risk and the audit trail	51
4.7	Creative, media, and communication work	51
4.8	Worked examples: the comparative method applied	52
4.9	Cross-domain synthesis	54
5	The AI-Native Enterprise	57
5.1	Beyond individual competence	58
5.2	Complementary capital and the deployment gap	58
5.3	Why most deployments fail	59
5.4	The operating model: platform, product, and practice	60
5.5	A maturity model	60
5.6	Role architecture	64
5.7	Accountability and the responsibility gap	66
5.8	Evaluation infrastructure in practice	67
5.9	The race between capability and infrastructure	68
5.10	Knowledge management and organisational learning	69
5.11	The economics of AI-native practice	69
5.12	Change management and labour relations	70
5.13	The cost of not building infrastructure	71
6	The Governance Regime	73
6.1	The governance operating system	73
6.2	From principles to practice	75
6.3	Disclosure	75
6.4	Confidentiality at the prompt layer	76
6.5	Contestability	77
6.6	Auditability	78
6.7	The accountability ladder	79
6.8	Bias, fairness, and the professional's obligation	80
6.9	Transparency and the explanation problem	81
6.10	Regulatory frameworks	81
6.11	The professional's governance checklist	82

7	Educating the AI-Native Professional	85
7.1	The inverted curriculum	85
7.2	What to teach: the five-capability model	86
7.3	How to teach: from recall to judgment	88
7.4	The assessment problem	89
7.5	The faculty challenge	91
7.6	Worked example: one course unit through the five capabilities	91
7.7	The competency portfolio	93
7.8	Continuing education and mid-career professionals	94
7.9	The institution's obligation	94
8	The Long Horizon	97
8.1	The frontier and its limits	97
8.2	The error frontier	98
8.3	The economic question	99
8.4	The substitution frontier	101
8.5	The agentic frontier	102
8.6	What we do not yet know	102
8.7	What to watch: indicators, not predictions	103
8.8	The durable strategy	104
8.9	The ten-year practice	104
8.10	The durable disposition	105
8.11	Closing: the professional's obligation	106
A	The AI-Native Capability Self-Assessment	109
A.1	Administration	109
A.2	Capability I: Problem Framing	109
A.3	Capability II: Delegation Design	110
A.4	Capability III: Verification	110
A.5	Capability IV: Context Engineering	110
A.6	Capability V: Ethical Stewardship	111
A.7	Interpretation	111
B	Verification Protocol Checklists	113
B.1	Checklist V1: Factual and Citation Verification	113
B.2	Checklist V2: Code and Analysis Verification	113
B.3	Checklist V3: Pre-Deployment Review for Delegated Agency	114
B.4	Checklist V4: Confidentiality and Data Handling	114
C	Governance Artifact Templates	117
C.1	Template G1: System Registration Record	117
C.2	Template G2: Risk Classification Worksheet	117
C.3	Template G3: Evaluation Report Skeleton	118

C.4	Template G4: Incident Record	119
C.5	Template G5: Disclosure Decision Matrix	119
D	Glossary	121
	Bibliography	123
	Index	139

The Paradigm Shift

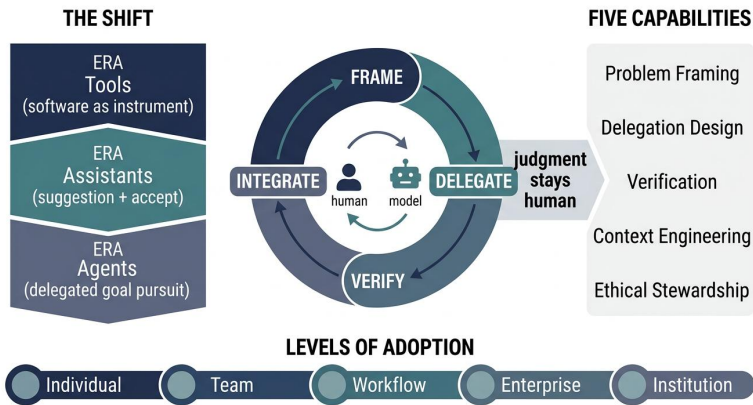


Figure 1.1. The AI-native paradigm shift: from instrument to interlocutor to agent, and the verification-centred practice that follows.

Chapter in Brief

Claim. Between 2022 and 2026, a class of system—the large, pre-trained, general-purpose model—became competent enough to be inserted into almost any knowledge-work process. This did not create a new tool; it created a new substrate. The professionals who thrive on that substrate are not the ones who master tools fastest but the ones who rebuild their practice around the constraint that now binds: verification.

Three distinctions. A *model* is a trained statistical system. A *system* is a model embedded in software with tools and memory. An *agent* is a system granted latitude to pursue a goal across multiple steps without step-by-step human authorisation. The distinction matters because each level multiplies the surface area for error.

What changed. Specialisation gave way to generality. The same system that drafts a contract can summarise a clinical trial, write SQL, and debug code. The economic consequence is not that one task became cheaper but that the fixed cost of entering a new task

collapsed.

What did not change. The professional's obligations: to be right, to show work, to protect confidences, to exercise judgment. A substrate that makes production nearly free makes verification the binding constraint.

1.1 From instrument to interlocutor: three eras of computational work

The relationship between professionals and computation has passed through three eras, and the boundaries between them explain what is genuinely new about the present one.

The first era was the era of the *instrument*. From the 1950s through the 2010s, computation was a tool that professionals picked up and put down: a spreadsheet, a database, a statistical package, a CAD system. The instrument executed procedures the professional specified in detail. It had no capacity to surprise in a way that mattered—a formula error or a data-entry mistake, not a fluent argument the professional had to evaluate. Expertise resided entirely in the professional; the instrument was an amplifier of procedures that were already understood (Agrawal et al., 2019; Bresnahan et al., 1999).

The second era, beginning in the mid-2010s, was the era of the *statistical interlocutor*. Deep learning made computation capable of pattern-based performance that no one had specified procedure by procedure. Image classifiers, speech transcribers, translation engines, and recommendation systems carried out recognition tasks that resisted procedural specification (Krizhevsky et al., 2017; LeCun et al., 2015; Silver et al., 2016). But these systems were narrow: a transcriber transcribed, a classifier classified. Each statistical system carried its own training corpus, its own engineering team, and its own boundary. The dominant pattern in professional automation remained specialisation—a system built for one narrow task, outperforming humans at that task and failing everywhere else. Each system required its own training data, its own integration, and its own switching cost (Agrawal et al., 2019; Bresnahan et al., 1999).

The third era, beginning in 2022, is the era of the *general-purpose model*. The transformer architecture and its scaling made a single pretrained system competent across reading, writing, reasoning, coding, and dialogue simultaneously (Brown et al., 2020; Devlin et al., 2018; Vaswani et al., 2025). A system trained on broad text corpora—then refined through instruction tuning and human feedback—could be inserted into almost any knowledge-work process without being purpose-built for it (Bommasani et al., 2021; Ouyang et al., 2022; Wei et al., 2022b).

From Symbolic AI to AI-Native Work, 1956-2026

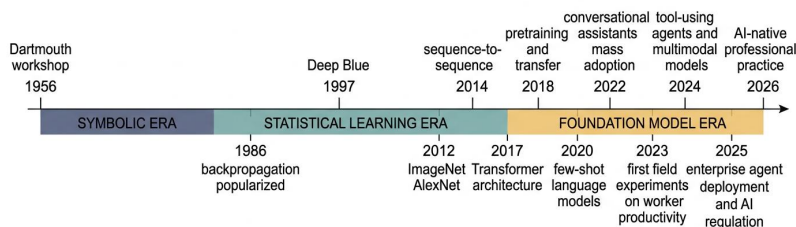


Figure 1.2. Milestones in the human–machine cognitive relationship, from procedural instruments to general-purpose models.

The timeline in Figure 1.2 compresses the three eras into one view. The pattern to notice is not the steady improvement of capability but the change in *kind*: from procedures specified by the professional, to patterns learned from the professional’s world, to a general competence that arrives complete and must be evaluated on the professional’s terms.

What changed between 2022 and 2026 is not that a new specialised system was built. What changed is that a *general-purpose model* became competent enough that it could be inserted into almost any knowledge-work process without being purpose-built for it (Brown et al., 2020; Bubeck et al., 2023). The same system that drafts a legal memo can summarise a clinical trial, write SQL queries, debug code, and generate a marketing plan. It is not better than the best specialised system at any single task. Its significance is that the *fixed cost* of entering a new task collapsed to roughly zero.

Table 1.1. Three eras of computational work compared.

	Era 1: Instrument	Era 2: Statistical system	Era 3: General-purpose model
Representative period	1950s–2010s	mid-2010s–2022	2022–
Performance basis	Specified procedures	Learned patterns	Pretrained general competence
Breadth	One procedure family	One recognition task	Language, code, reasoning, dialogue
Basis of trust	Deterministic execution	Measured accuracy	Case-by-case verification
Professional’s role	Specify and operate	Train, integrate, monitor	Frame, delegate, verify, govern
Typical failure	Formula or data error	Boundary error	Fluent, confident error

1.2 What makes a general-purpose model different

Three technical properties distinguish the general-purpose model from every previous class of professional automation, and each has a direct professional consequence.

Scaling. Performance across a broad frontier of tasks improves predictably with model size, data, and compute (Hoffmann et al., 2022; Kaplan et al., 2020). The professional consequence is that capability arrives as a completed fact: a new model version changes the error profile of a whole practice area at once, without any local decision to adopt it. Competence is something the professional receives and must evaluate, not something the organisation builds incrementally.

Pretraining and in-context learning. A pretrained model learns from the statistical structure of the professional’s world—the language of contracts, clinical notes, and code—and then adapts within a conversation, without further training, to the task at hand (Bommasani et al., 2021; Brown et al., 2020). The professional consequence is that the model carries prior assumptions from its training corpus that the professional did not choose and cannot fully inspect: it reproduces the form of a cited statistic without having the content, and it inherits the biases of the text it was trained on (Bender et al., 2021; Weidinger et al., 2022). Retrieval-augmented generation grounds outputs in documents the professional supplies—and makes the quality of that grounding a professional responsibility rather than a technical one (Lewis et al., 2020).

Instruction tuning and feedback. Refinement through human feedback makes the model responsive to instructions stated in natural language, including professional specifications of format, tone, and scope (Bai et al., 2022; Ouyang et al., 2022; Ziegler et al., 2024). The professional consequence is delegation: the model responds to the specification the professional writes. The quality of the specification—its precision about what “correct” means, its explicitness about exclusions—now determines the quality of the output to a degree that no prompt trick can substitute for. Specification, as Chapter 2 develops, is a professional skill.

What these three properties produce together is generality: the same system at the boundary of many professional domains at once (Bommasani et al., 2021; Bubeck et al., 2023). The systems of the second era could surprise only at their own boundary. The general-purpose model can surprise—fluent, confident, plausible—everywhere.

1.3 Models, systems, and agents

The preface distinguished three terms that will recur throughout this book. The distinction is not academic; it tracks three different failure modes and

three different governance obligations.

A *model* is a trained statistical system. Given an input, it produces an output. It has no persistent state, no tools, and no ability to act on the world. Its failure mode is *inaccuracy*: the output may be fluent, confident, and wrong (Hendrycks et al., 2021; Ji et al., 2023). The professional's obligation when using a model is *verification*: reading the output, checking it against sources, and deciding whether to rely on it.

A *system* is a model embedded in software that gives it tools (search, code execution, database access) and memory (conversation history, retrieved documents, persistent state). Its failure mode is *composition*: the model may use a tool incorrectly, retrieve the wrong document, or combine accurate components into an inaccurate whole (Schick et al., 2023; Wu et al., 2023). The professional's obligation when using a system is to inspect the *trace*—not just the final output but the intermediate steps that produced it.

An *agent* is a system granted latitude to pursue a goal across multiple steps without step-by-step human authorisation. Its defining property is *autonomy*: the agent decides which action to take next, which tool to call, and when to stop (Shinn et al., 2023; Yao et al., 2022). Its failure mode is *compounding error*: a small mistake early in a long trajectory—a misread number, a wrong retrieval, a plausible but incorrect inference—propagates through subsequent steps and becomes difficult to detect in the final output (Magesh et al., 2025). The professional's obligation when overseeing an agent is to set the goal, the boundary, and the stop condition before the trajectory begins, and to audit the full trace afterward. Agents that plan, call tools, and self-correct are no longer hypothetical: they are the systems arriving in research, software engineering, and office practice now (Boiko et al., 2023; Chen et al., 2021; Jimenez et al., 2023; Wang et al., 2023).

The autonomy trap

common mistake is to treat an agent as if it were a model—checking only the final output and ignoring the trajectory. This works when the trajectory is short and the task is well-specified. It fails catastrophically when the trajectory is long, the task is ambiguous, and the agent has access to consequential tools (sending emails, executing transactions, modifying shared documents). The longer the trajectory and the more consequential the tools, the less the final output tells you about whether the agent did the right thing. Agent evaluation methodology lags deployment for exactly this reason (Kapoor et al., 2024; Ruan et al., 2023).

The distinction matters enough that Section 1—this section—is devoted to it. Throughout the book, the word *model* refers to a trained statistical system, *system* to a model embedded in software with tools and memory, and *agent* to a system granted latitude to pursue a goal across multiple steps without step-by-step human authorisation.

1.4 The economics of a new substrate

When the cost of producing a first draft, a summary, or a code snippet falls to near zero, the binding constraint shifts. It is no longer “can we produce enough?” but “can we verify fast enough?” (Brynjolfsson et al., 2023; Handa et al., 2025)

Noy and Zhang (2023) conducted one of the first controlled experiments on this shift. They assigned knowledge workers to complete writing tasks with and without access to a general-purpose model. The model-assisted group completed tasks 37% faster and produced output that independent raters scored 18% higher in quality. The gain was largest for participants whose baseline skill was lowest—a compression effect that has been replicated in subsequent studies (Dell’Acqua et al., 2026; Peng et al., 2023).

But the same studies found a second pattern. When participants were asked to judge whether their model-assisted output was correct, their accuracy was only marginally higher than chance. The model made the production of plausible text nearly free, but the *evaluation* of that text did not become free. If anything, it became harder, because the output was more fluent and more confident than anything the participant could have produced unaided, which made errors harder to detect (Cemri et al., 2025; Cui et al., 2026).

The evidence base is now broad enough to state the field-experiment pattern precisely, and Table 1.2 organises it. Three findings recur across settings:

1. **Speed gains are robust; quality gains are conditional.** Speed improvements appear in nearly every setting. Quality improvements appear where the task has a verifiable ground truth or where the professional’s specification is precise (Dell’Acqua et al., 2026; Peng et al., 2023).
2. **Compression is real but bounded.** The largest measured gains accrue to the lowest-skill performers, compressing the skill gradient within and between roles (Dell’Acqua et al., 2026; Noy and Zhang, 2023). But in tasks with a high verification cost—where the professional must check claims against sources the model cannot access—experienced practitioners retain an advantage precisely because their verification is faster and more reliable (Deng et al., 2025; Otis et al., 2024).

3. **Heterogeneity of distribution is the key field finding.** Field deployments report effects that range from strongly positive to flat to negative, and the differences track task design—how the work was decomposed, what was verified, and who was accountable—rather than model version (Beane, 2019; Brynjolfsson et al., 2023).

Table 1.2. The field-experiment evidence pattern, as it stood in mid-2026. Reported effects are the studies’ own headline findings; the professional implication is this book’s reading.

Study	Setting and headline effect	Professional implication
Noy and Zhang (2023)	Mid-level writing tasks; 37% faster, 18% higher quality	Production is no longer the bottleneck; evaluation is
Peng et al. (2023)	Software tasks; 55% faster completion	Verification via tests replaces verification via reading
Dell’Acqua et al. (2026)	Consulting tasks inside the capability frontier; +40% quality, +12% tasks	Gains are largest inside, not at, the model’s frontier
Dell’Acqua et al. (2026)	Consulting tasks outside the frontier; 19%–24% more likely to produce wrong answers	The frontier is invisible from fluency alone; only domain expertise sees across it
Cui et al. (2026)	Idea generation; higher novelty, more homogeneous ideas	Volume rises; differentiation requires the professional’s judgment
Brynjolfsson et al. (2023)	Customer support; +14% productivity, +34% for novices	Compression within roles; skill complementarity shifts to handling exceptions
Otis et al. (2024)	Professional writing with verification demands; flat to negative for experts	Where verification dominates production, expertise retains its advantage
Noy and Zhang (2023)	Self-judgment of model-assisted output; accuracy near chance	Reliance must be calibrated, not assumed (Chapter 2)

This is the central economic fact of the new substrate. It does not eliminate the need for expertise; it changes what expertise is *for*. The expert’s value shifts from producing the right answer to knowing whether a given answer is right—and to designing the process by which that judgment is made

cheaply, repeatedly, and at scale. Acemoglu and Restrepo (2019) made the complementary-investment point for the automation wave of the 2010s: productivity gains from automation materialise only where organisations invest in the tasks and structures that surround the technology. Autor (2015) gave the durable formulation: expertise complements automation where the work involves flexibility, judgment, and common sense—and the arrival of general-purpose models moved that boundary without erasing it (Autor, 2015; Hemmer et al., 2023; Vaccaro et al., 2024).

The consultant's dilemma

management consultancy deployed a general-purpose model to produce first drafts of market analyses. Consultants reviewed the drafts and added analysis. Productivity rose 30% on the first pilot. But when a senior partner spot-checked ten drafts against primary sources, six contained at least one fabricated statistic—a number that looked real, was formatted correctly, and was entirely invented. The model had learned the *form* of a cited statistic without having the *content*. The firm's response was not to stop using the model but to add a verification step: every number in every draft had to be traced to a source before the draft left the building. The cost of production fell; the cost of verification—and the cost of getting verification wrong—rose. The pattern is not hypothetical: the inside-frontier/outside-frontier divergence measured by Dell'Acqua et al. (2026) is the measured form of exactly this case.

The substrate audit

nce a quarter, perform the audit against this chapter's structure:

1. List every task you delegate, with its substrate class (model, system, agent) and its placement on the delegation ladder (Chapter 3).
2. For each, name the failure mode you are accepting (inaccuracy, composition, compounding error) and the verification instrument that targets it.
3. Mark which placements you designed and which you drifted into. The drifted placements are the quarter's design work.

The audit prices the drift: the outside-frontier case (Section 1) grows exactly where placements drift, because fluency makes the drift invisible until the spot-check surfaces it (Dell'Acqua et al., 2026; Kapoor et al., 2024).

1.5 Augmentation, automation, and the verifiability frontier

The public debate frames the question as augmentation versus automation: does the model assist the professional or replace them? The framing is useful for policy and unhelpful for practice, because it treats the boundary as a property of the technology. In deployment, the boundary is a property of the *task design*: of how the work is decomposed, what is verified, and who is accountable for the verification.

The useful construct is the *verifiability frontier*. For a given task in a given practice, there is a frontier between outputs the professional can verify within the time the task’s stakes warrant and outputs they cannot. Model use inside the frontier—the professional can check the output—behaves like augmentation, and the measured productivity gains are large. Model use outside the frontier—the output cannot be checked—behaves like blind automation, and the measured error rates rise exactly when fluency is most convincing (Dell’Acqua et al., 2026; Ji et al., 2023). The frontier is not fixed: verification discipline (Chapter 3) moves it, and trust without verification erodes it.

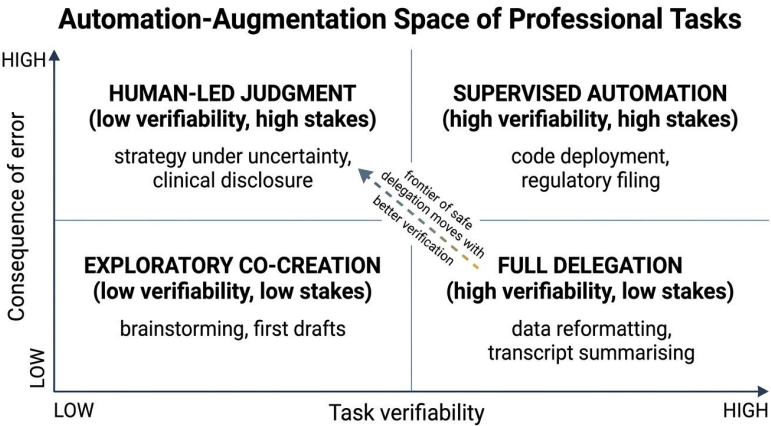


Figure 1.3. The automation–augmentation matrix: task exposure against verifiability. The productive quadrant is high exposure with high verification—not low exposure.

Table 1.1 organised the eras; Figure 1.3 organises the frontier. The exposure question—what share of a task’s activities a model can do—is the one Eloundou et al. (2024) answered at the occupation level and Grace et al. (2025) at the time-horizon level. But exposure alone predicts nothing: exposure without verification is the outside-frontier case, and the outside-frontier case is where fluent errors survive. The quadrant to design for is high exposure with high verification. That design is the subject of this book.

Two research lines sharpen the frontier from the system side. Calibration work asks whether a model’s confidence tracks its accuracy—and finds that unsupervised fluency does not, which is why the professional’s own verification cannot be delegated to the system’s self-report (Gerlich, 2025; Kapoor et al., 2024). Evaluation work asks whether today’s benchmarks measure the properties professionals need—and finds systematic gaps between benchmark accuracy and reliable, specification-following, audit-ready performance (Chen et al., 2021; Chollet, 2019; Jimenez et al., 2023; Kapoor et al., 2024). Both lines converge on the same practice conclusion: the guarantee the professional needs is constructed by the professional’s process, not inherited from the model’s average behaviour.

1.6 What “AI-native” means (and does not mean)

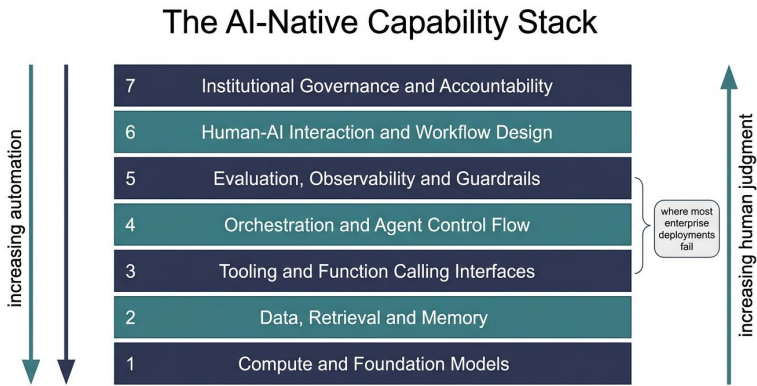


Figure 1.4. The AI-native capability stack: five capabilities resting on the professional’s existing expertise, with verification as the load-bearing layer.

AI-native names a practice, not a toolset. The AI-native professional has rebuilt their working method around the new cost structure: specification, delegation, verification, governance, and adaptation (Figure 1.4). The five capabilities are the organising scheme of this book—Chapter 2 develops them individually, Chapter 3 operationalises them in task design, Chapter 4 adapts them to specific professions, Chapter 5 scales them to organisations, Chapter 6 grounds them in governance, and Chapter 7 teaches them.

The term does *not* mean:

- **Adoption of every tool.** Tool churn is a constant of the period (McDuff et al., 2025; Merchant et al., 2023). The AI-native practice is defined by its method, not by the current system names; the method survives tool changes because its load-bearing layer is verification, which does not churn.

- **Maximal delegation.** Delegation is a calibrated decision, and the calibration is domain- and stakes-dependent (Chapter 3). Maximal delegation without verification is the outside-frontier case, and its measured error rates are the argument against it.
- **Abandonment of expertise.** Offloading without engagement erodes the very skill that makes offloading safe—the metacognitive and domain knowledge needed to detect error (Gerlich, 2025; Kelly and Risko, 2026; Long and Magerko, 2020). The AI-native professional uses the model to go further into the domain, not to stop going into it.

What the term *does* mean is that the professional’s practice works *with* the substrate’s grain: production is cheap, so specification and verification are where value concentrates; fluency is unconditional, so trust is conditional on evidence; capability is received as a fact, so adaptation is a professional routine rather than an organisational exception (Brynjolfsson, 2022; Rahwan et al., 2019).

1.7 The argument of the book

The book’s argument proceeds in five moves.

First, the new substrate changes the economics of professional work—production nearly free, verification the binding constraint—and the professionals who thrive are the ones who rebuild practice around that constraint (Chapter 1).

Second, the rebuilding is specified in five learnable capabilities—problem framing, delegation design, verification, context engineering, and ethical stewardship—with calibrated reliance and explicit cognitive costs (Chapter 2).

Third, the capabilities are operationalised in task and workflow design: the delegation ladder, verification architecture, and teaming patterns that make the verifiability frontier a designed object rather than an accident (Chapter 3).

Fourth, the same capability set adapts across professions with different evidence standards and regulatory demands—software, medicine, law, research, finance, and creative work—and scales from individual practice to organisational structure and governance (Chapter 4, Chapter 5, Chapter 6).

Fifth, the capabilities are teachable, assessable, and durable under capability change—and under every scenario for the long run, the professional’s own verification-centered practice is the robust strategy (Chapter 7, Chapter 8).

1. Identify three tasks in your current work that you delegate to a model, a system, or an agent. For each, name the substrate class (model, system, agent), the specific failure mode (inaccuracy, composition, compounding error) that concerns you most, and the verification step you currently perform.
2. Choose a model-assisted output you produced this week. Without re-reading the output, write down what you would check if you had to stake your professional reputation on its accuracy. Then check it. How many errors did you find?
3. Place your three tasks on the automation–augmentation matrix (Figure 1.3): estimate the model’s exposure on each task and your own ability to verify the output within the time the stakes warrant. Which quadrant does each sit in—and is that the quadrant you designed, or the one you drifted into?
4. For the next task you delegate to a model, write a one-sentence specification of what “correct” means before you produce the output. Compare your specification to the output. Did the output meet it?
5. Find the inside-frontier and outside-frontier cases in your own work (Section 1): one task you verify within the stakes’ time and one you cannot. For the outside-frontier case, name the containment cost you are accepting and who bears it if the error survives.
6. Run the field-experiment pattern against yourself: for the task from Exercise 4, record your completion time and your judgment accuracy for two weeks (with and without the model). What did the recorded heterogeneity show that the headline feeling did not?

Further Reading

- Bubeck et al. (2023)—the paper that catalysed the “GPT-4 is general-purpose” claim, with careful discussion of its limitations.
- Bommasani et al. (2021)—the systematic statement of the foundation-model paradigm and its open questions.
- Brynjolfsson et al. (2023)—the most thorough productivity study to date; read the methodology, not just the headline.
- Dell’Acqua et al. (2026)—the inside-frontier/outside-frontier divergence that defines the verifiability frontier.
- Eloundou et al. (2024)—a systematic assessment of which occupations are most exposed to general-purpose models and why.

- Acemoglu and Restrepo (2018)—the framing of the race between education and technology that structures the argument of this book.

