COMPANION VOLUME

Teaching AI in the Workplace

Eight modules, lesson plans, drills and everything needed to run it yourself

INCOPREA RESEARCH · EDITION 1.0

4 November 2026

Teaching AI in the Workplace

Eight modules, lesson plans, drills
and everything needed to run it yourself

Incoprea Research

Edition 1.0 · 4 November 2026

How to use this workbook

This is not a book about AI. It is a set of materials for a person who has been asked to "do something about AI" in an organisation and now has to stand at the front of a room.

Everything here is written so that it can be used three ways: read straight through as a primer, followed as a lesson plan in the room, or handed out section by section as participant material. The drills are written to be run, not discussed.

It assumes the audience is competent at their job and nervous about this one. That combination is normal and it governs the design: **no jargon without a plain definition, no demonstration without a failure case, and nothing that requires an account, a licence or a budget.**

> **The core promise you make to the room:** "You will leave able to do three things with these tools tomorrow, and able to tell when they are lying to you. The second skill is the one that keeps you out of trouble."

What is in the box

Section	What it is	Time

Modules 1–8	Full lesson plans with drills	45–90 min each
Run sheets	Half-day, one-day and two-day agendas	—
Handouts A–E	Printable participant material	—
Assessment	Rubric and ten-question check	20 min
Sales copy	Course description, pricing, objection handling	—
Facilitator notes	What goes wrong, and what to do	—

Before the session

Send this to participants two days ahead. It halves the time lost to orientation and it changes the room: people arrive with a real task instead of a general interest.

Participant pre-work (15 minutes)

1. Bring **one real document** from your own work — something long that you would normally have to summarise or rewrite. Redact client names; we do not need them.
2. Bring **one task** that takes you longer than it should.
3. Write down **one thing the tool has got wrong** for you, or tell us you have not used it yet. Both answers are useful and neither is judged.

> **Why the third question matters:** the room divides into people who trust the tool and people who refuse it. Both groups are usually wrong in the same way — neither has a rule for when it is safe. Starting from a personal failure shortens that argument by an hour.

Module 1 — What these tools are and are not

Purpose: replace mystique with a working model, in twenty minutes, without overselling or dismissing.

Plan (60 minutes)

1. Open with a question, not a slide: "What has it done well for you, and what has it got confidently wrong?" Collect both columns. The room usually has both, and the wrong answers are the more instructive.
2. Explain the mechanism in one sentence: *it predicts the next word, extremely well, having read an enormous amount.* That single sentence explains both the fluency and the confabulation, and it is enough. Do not teach architecture.
3. Demonstrate a failure on purpose. Ask for a citation, a date, or a statistic in a domain the room knows. The wrong answer arrives formatted exactly like a right one. **This is the moment the workshop exists for.**
4. State the division of labour: the tool generates, you verify. Nothing in the room is going to change that.

Drill 1.1 — The confidence test. Pairs, eight minutes. Each person asks the tool for five factual claims in their own specialism, then checks all five against something real. Tally how many were wrong and how many *sounded* wrong. The gap between those two numbers is the lesson.

> **What goes wrong:** someone insists "but it's improving". Agree, then redirect: "It is. The checking requirement hasn't moved, because it never depended on the tool being bad."

Module 2 — The three jobs it does well

Purpose: give people something to *use* before lunch, so the day isn't only warnings.

The three, and why they are safe

Job	Why it works	Example in a normal office
Drafting	You review the output anyway; the cost of an error is one edit	First draft of an email, memo, policy summary
Transforming	The input is in front of you, so the output is checkable by comparison	Long document → bullet summary; notes → minutes; table → paragraph
Explaining	You will notice if the explanation is wrong, because it is about your own work	"Explain this clause", "what does this metric measure", "what am I missing"

Drill 2.1 — Real work, real deadline. Each participant brings one genuine task from their desk. Fifteen minutes to produce a draft with the tool, then five minutes rewriting it in their own voice. The rewrite is compulsory: it is where the value and the responsibility both live.

Facilitator note: people who arrive sceptical usually leave converted by *transforming*, not by drafting. Summarising their own 40-page document is the demo that lands.

Module 3 — Where it fails, and how you notice

Purpose: build a reflex, not a policy.

The four failures, in the order you will meet them

1. **Confabulation.** A confident, formatted, false statement — most often a citation, a number, a date or a person.
2. **Agreement.** It adopts the premise of your question. Ask "why is this policy bad?" and it will explain why, without noting that it might be good.
3. **Drift.** On long tasks the output quietly changes premise halfway. The end looks like the beginning but is no longer answering the same question.
4. **Plausible arithmetic.** Numbers that are internally consistent, correctly formatted, and wrong — because the inputs were wrong or the operation was invented.

Drill 3.1 — Break it deliberately. Groups of three, twelve minutes. Each group must produce one convincing falsehood in their own domain — something a colleague would believe — and then explain how they would catch it. **The catching is the deliverable.**

> **The reflex to install:** before any output leaves your desk, name the check you performed. If you cannot name it, you have not checked.

Module 4 — Asking for the right thing

Purpose: replace "prompt engineering" folklore with three habits that work.

The three habits

Give the audience and the purpose. "Explain X to a new starter who needs to do Y by Friday" outperforms "explain X" every time.

Give the material, not a description of it. Paste the document. Summaries of summaries are where invention enters.

Ask for the test, not the answer. "What would show this is wrong?" is the single most useful question to ask a machine, and almost nobody asks it.

> **Drill 4.1 — Three drafts of one request.** Each person writes the same request three ways: bare, with audience and purpose, and with material attached. Compare outputs. The improvement is usually larger than people expect, and it teaches that specificity beats cleverness.

Why these three and not a framework

Every popular prompting framework is a re-description of the same three moves: say who it is for, give it the material, ask for a test. Frameworks age because they are named after products that change; the moves do not.

Request	What it usually returns	What to add
"Summarise this"	A generic abstract	The reader and what they must decide
"Write an email about X"	Polite emptiness	The relationship, the ask, the consequence of refusal
"What is wrong with this plan?"	Agreement with your premise	"Assume it is wrong; list three reasons it fails"

"Give me options"	Five variations of one idea	A constraint that forces genuine difference

> **The uncomfortable rule:** if you cannot state the check you will run on the answer, you are not ready to ask the question. Asking well and checking well are the same skill seen from two ends.

> **Facilitator note:** resist the urge to teach a prompt framework. Frameworks date; the three habits do not.

Module 5 — Checking without becoming the bottleneck

Purpose: the practical objection — "if I have to check everything, this is slower".

The sorting rule

Sort outputs by **what happens if it is wrong**, not by how good it looks.

If wrong…	Check	Typical use
Nothing much	Skim	Brainstormed titles, tone alternatives
Wasted effort	Read against source	Summaries, reformatting
Reputational	Verify facts and figures	Anything sent externally
Irreversible	Do not use the tool	Legal, medical, safety, money moving

Drill 5.1 — Sort your own week. Participants list eight tasks from last week and place each in a row. Ten minutes. Most discover two tasks that should have been in the bottom row and were not.

Module 6 — Writing it down: the team's rules

Purpose: convert personal habits into something a team can hold each other to.

A one-page policy that actually gets used

1. **What we use it for** — three things, named.
2. **What we never put in it** — client data, personal data, unannounced financials.
3. **What must be checked, and by whom** — one line per category.
4. **What we never send without a human author reading it end to end**.
5. **How we record that we used it** — a sentence, not a system.

> **Drill 6.1 — Draft the policy.** In groups, thirty minutes, using the five headings. Then swap with another group and try to break theirs. Policies that survive that exchange are the ones that get used.

> **Common failure:** a policy that bans the tools. It is ignored within a fortnight and teaches the team that the rules are decorative. Better to permit narrowly and enforce visibly.

The policy that gets written but never used

Most policies fail for one of three reasons, and each has a fix that costs nothing.

Why it fails	What it looks like	Fix
Too long	Four pages nobody has read	One page, five headings, printed and pinned up
Unenforceable	"Staff must not use AI for sensitive work" — with no definition	Name the categories, not the principle
No owner	Written by a committee, revised by nobody	One named person, one review date, in the header

The awkward cases, answered in advance

1. **"Can I put a client's document in?"** Only if the client's contract permits it. Absent a clause, the answer is no, and the clause is worth requesting.
2. **"Is my draft confidential?"** Treat anything typed into these tools as leaving the building. If that is unacceptable, the tool is not the right one for that task.
3. **"Who is responsible if it is wrong?"** The person who sent it. This is the only answer that survives contact with a regulator or a client.
4. **"Can I use it to decide about a person?"** No. Screening, ranking and evaluation of individuals carry legal exposure and belong nowhere near a generative tool.
5. **"What about the recording of the meeting?"** Transcription is a separate decision from generation, and the consent question belongs to the people in the room.

Module 7 — Building one small thing

Purpose: move from consumer to builder, at the smallest possible scale.

What counts as a thing

A checklist, a template, a spreadsheet formula, a script, a saved prompt with instructions — anything that makes the good behaviour easier than the bad behaviour.

Drill 7.1 — Build in ninety minutes. Each participant builds one artefact that a colleague could use without them. Requirements: it must fit on one page, it must say what it does *not* cover, and it must be handed to someone else before the end of the day.

Facilitator note: this module is where the confident participants separate from the anxious ones. Pair them deliberately. The anxious person often builds the more usable artefact, because they think about failure.

Module 8 — Teaching it onward

Purpose: make the workshop reproduce, so the organisation does not need you again.

The three things a colleague needs

1. **One demonstration of failure** — they must see a confident wrong answer in their own domain.
2. **One quick win** — a task they will do this week.
3. **One rule** — the check they must run before sending.

> **Drill 8.1 — Teach it in five minutes.** Each participant delivers all three to a partner, timed. Then swap. What survives at five minutes is what you can actually transfer.

Run sheets

Half day (3.5 hours)

Time	Module	Note
0:00	1 — What it is and is not	Include the deliberate failure
0:50	2 — Three jobs	Bring-your-own-task drill
1:50	Break (15)	—
2:05	3 — Where it fails	Break-it drill is the core
2:50	5 — Checking	The sorting rule
3:20	Close: three commitments	Each person names one

One day (6 hours)

Half-day blocks plus Modules 4 and 6. The policy drill needs the extra time and is worth it: a team that leaves with a one-page policy has something to show for the day.

Two days

Day one: Modules 1–5. Day two: Module 6 in the morning, Module 7 after lunch, Module 8 to close. Day two is where people build something real, so protect the ninety minutes.

Slide outlines

Ten bullets per module — enough to build a deck in an hour, deliberately not a finished deck. Resist adding slides; the drills are the course.

Module 1 — What it is and is not

1. Two questions: what has it done well, what has it got confidently wrong
2. One sentence on the mechanism: predicts the next word, having read a great deal
3. That explains both the fluency and the invention
4. Live failure demonstration in your domain
5. It formats wrong answers exactly like right ones
6. Why it cannot know what is true
7. It has read everything and checked nothing
8. The division of labour: it generates, you verify
9. This does not change when the models improve
0. Drill 1.1 — the confidence test

Module 2 — The three jobs

1. Drafting — you review anyway
2. Transforming — compare against the source
3. Explaining — you notice errors in your own field
4. Why these three are safe: the check is built into the task
5. What is not safe: anything you cannot check
6. A real document, summarised live
7. The rewrite is compulsory — that is where responsibility lives

8. Bring your own task
9. Fifteen minutes producing, five rewriting
0. Drill 2.1

Module 3 — Where it fails

1. Confabulation: formatted, confident, false
2. Agreement: it adopts your premise
3. Drift: the premise changes mid-task
4. Plausible arithmetic: consistent and wrong
5. What each looks like in an ordinary document
6. The symptom to watch for in each case
7. Why you cannot check everything
8. Sorting by consequence instead
9. Drill 3.1 — manufacture a convincing falsehood
0. The deliverable is the catching, not the falsehood

Module 4 — Asking for the right thing

1. Three habits, not a framework
2. Give the audience and the purpose
3. Give the material, not a description of it
4. Ask for the test, not the answer
5. Why frameworks date and moves do not
6. Four common requests and what they return
7. The uncomfortable rule: no check, no question
8. Drill 4.1 — three drafts of one request
9. Compare the outputs side by side
0. Specificity beats cleverness

Module 5 — Checking

1. The objection: "if I check everything this is slower"
2. Sort by consequence, not by appearance
3. Four rows: nothing much, wasted effort, reputational, irreversible
4. What each row permits
5. Most teams have two tasks in the wrong row
6. The check must be nameable
7. If you cannot name the check, you have not checked
8. Drill 5.1 — sort your own week
9. Which task surprised you
0. What changes tomorrow

Module 6 — The team's rules

1. Five headings on one page
2. What we use it for, named
3. What never goes in
4. What is checked, and by whom
5. What a human must read in full
6. How we record that we used it
7. Why bans fail within a fortnight
8. Permission narrowly, enforce visibly
9. Drill 6.1 — draft it, then try to break another group's
0. The awkward cases, answered in advance

Module 7 — Building one small thing

1. What counts as a thing: checklist, template, formula, saved instruction
2. The test: a colleague uses it without you

3. One page maximum

4. It must state what it does not cover

5. Ninety minutes, no scope creep

6. Hand it to someone before the end of the day

7. The anxious build better artefacts than the confident

8. Drill 7.1

9. Show two artefacts to the room

0. Why this module converts sceptics

Module 8 — Teaching it onward

1. Reproduction is the goal

2. One failure demonstration

3. One quick win

4. One rule

5. Five minutes, timed

6. What survives five minutes is what transfers

7. Drill 8.1

8. Pair a confident person with an anxious one

9. Ask each pair what they dropped

0. Close: three commitments from the room

Run it in 90 minutes

A single session for people who want to try this before committing to a day. It is not a condensed version of the eight modules — it is the one hour and a half that changes behaviour, with everything that can be cut, cut.

Min	What happens	Why it stays in
0–5	Two questions to the room: what has it done well, what has it got confidently wrong.	Establishes that both happen, without a lecture.
5–15	One sentence on mechanism. Then the deliberate failure: ask for a citation or number in their own domain and check it live.	The whole session exists for this moment.
15–40	**Drill: their own document.** Each person summarises a real document, then rewrites the summary in their own words.	A quick win with a built-in check; the rewrite is the responsibility.
40–55	Three failures fast: confabulation, agreement, drift. One example each, from their answers.	Names what they have already seen.
55–75	**Drill: sort your week.** Eight real tasks into four consequence rows.	Produces a usable rule, not a feeling.
75–85	The four rows as a handout. Each person writes the one check they will run.	Leaves with a commitment in writing.
85–90	Close: one sentence each on what they will stop doing.	Stopping is the behaviour change; starting is the easy part.

What to cut, in this order

1. **Everything about prompting frameworks.** They date, and the drill teaches the same thing faster.
2. **The policy module.** It needs an hour and a real argument; 90 minutes cannot hold it. Mention that a policy is the next step and stop.
3. **The build module.** Ninety minutes is not enough to build something someone else will use, and a rushed artefact is worse than none.

4. **Your own examples.** Use theirs. Every minute spent on a story about your work is a minute not spent on theirs.

What you need

One machine with a working account · a projector · printed copies of Handout C (the sorting rule) · one real document per person, brought by them · a timer. Nothing else. No slides are strictly required; the two questions and a live failure will carry the first fifteen minutes.

> **Success criterion:** not "they enjoyed it". At the end, every person can name one check they will run and one thing they will stop doing. If they cannot, the session did not take — and the most common cause is that you talked for the first forty minutes instead of the first fifteen.

> **If you have only 45 minutes:** keep the failure demonstration, the sort-your-week drill, and the written check. Drop everything else. That is the irreducible core, and it still works.

Handouts

Handout A — The three jobs

Drafting (you review anyway) · Transforming (compare against source) · Explaining (you notice errors in your own field). If a task is none of these, slow down.

Handout B — The four failures

Confabulation · Agreement · Drift · Plausible arithmetic. For each: the symptom and the check.

Handout C — The sorting rule

If wrong: nothing much → skim. Wasted effort → read against source. Reputational → verify facts. Irreversible → do not use.

Handout D — The team policy template

Five headings: what we use it for; what never goes in; what is checked and by whom; what a human must read in full; how we record use.

Handout E — The five-minute handover

One failure demonstration, one quick win, one rule. That is the entire curriculum for a colleague.

Assessment

Criterion	Not yet	Competent	Can teach it
Recognising confabulation	Trusts output	Checks facts	Demonstrates the failure to others
Choosing tasks	Uses for everything	Sorts by consequence	Explains the sorting rule
Checking	Skims	Reads against source	Designs the check for a team
Policy	None	Follows one	Writes and defends one
Building	Consumes	Has one artefact	Has one others use

Ten-question check: two on mechanism, three on failures, three on the sorting rule, one on policy, one on the handover. Pass mark is seven, and question nine should be "name the check you ran".

Running it as a business

Course description (use verbatim)

Most AI training teaches people to write prompts. This teaches them to use these tools without discrediting their own work. In one day, participants learn the three jobs the tools do well, the four ways they fail, and a one-page rule for the team. Everyone leaves with a working artefact and a check they can actually run.

Pricing shapes

Format	Typical range	Sells on
Open workshop, per seat	$150–400	Individual confidence
In-house half day	$1,200–2,500	Team alignment
In-house full day	$2,500–5,000	The policy they leave with
Two-day with build	$5,000–9,000	A working artefact
Online, per seat (2 × 90 min)	$90–200	Convenience; lowest delivery cost
Online, in-house (2 × 90 min)	$800–1,500	Team alignment without travel

> **Price the outcome, not the hours.** A team that leaves with a policy it will actually use is worth more than the eight hours you spent, and the buyer knows it.

Objections and answers

They say	You say
"Our staff already use it."	"Then the useful session is the one on checking, not the one on starting."
"Can't we just watch a video?"	"A video can't produce your team's policy or test whether they'd catch a wrong number in your own data."
"It'll be out of date in a year."	"The tools will change. The four failure modes have not, and that's the part people take away."

"We need something more advanced."	"Most advanced failures are basic ones at scale. We start where the damage is."

Facilitator notes

The five things that go wrong

1. **You over-demonstrate.** Keep the tool on screen for minutes, not hours. The drills teach more than the demo.
2. **One person dominates.** Use pairs for every drill; it halves the airtime problem.
3. **The room splits into zealots and refusers.** Name the split out loud. Both are usually wrong in the same way: the zealot has not been burned yet.
4. **Someone has a compliance horror story.** Let it land, then ask what check would have caught it. It becomes your best material.
5. **You get asked to predict the future.** Decline. Say what does not change: the need to check, and the responsibility of the person who sends it.

Materials checklist

Projector and one machine with a working account · printed handouts A–E · one real document from the client to summarise · sticky notes · a timer · a spare laptop in case the demo fails (it will, and that is teachable).

After the session

The workshop decays in a fortnight unless something is left behind. Two artefacts prevent that: a follow-up message that asks for evidence, and a fixed date to look at it.

Follow-up email (send within 24 hours)

Subject: Your one check, and one request

Thank you for today. Two things.

First: the check you chose is written below. If you have not chosen one, reply with the sentence "my check is ___" and I will help you make it runnable.

Second: a request. In two weeks I will ask whether you used it. If you did, tell me what it caught. If you did not, tell me what got in the way — that is more useful to me than a compliment, and it is how the next version of this course gets better.

[Name]

Thirty-day check-in (three questions)

1. What did the tool get wrong that you caught?
2. What did it get wrong that you sent anyway?
3. What have you stopped using it for, and why?

> **Why question two matters most.** Self-reported success is unreliable; a confession of the near-miss is the only evidence that the checking habit actually formed. If nobody has a near-miss to report, they are either not using the tool or not checking it.

Credential wording, if you offer one

Keep it descriptive and unfalsifiable-by-implication: *"Completed one day of practical training in the use and validation of generative tools, including failure recognition and a written team policy."* Avoid the word certified unless an awarding body is behind it, and never imply a qualification you do not hold.

Colophon

Teaching AI in the Workplace: A Trainer's Workbook. First edition, 4 November 2026. Companion volume in the Incoprea research series, uniform with *Discovery with Machines*; the two share a method and can be used together — the first for the researcher working alone, the second for the person who has to teach a room.

Written as a single self-contained HTML document and converted to PDF and EPUB from the same source. Text CC BY 4.0: copy it, run it, charge for your time, and give the material away.