



Prompting and context design for business teams

Writing requests, supplying context and building reusable instructions that make outputs dependable.

WORKS WITH THE ASSISTANTS TEAMS ALREADY USE



EDITION 40 / EMBER & AMBER

EXECUTIVE WORKING EDITION - 08 OCT 2026

40

AT A GLANCE

Writing requests, supplying context and building reusable instructions that make outputs dependable.



Most disappointing AI output comes from an under-specified request, not from a weak model. This playbook treats prompting as a working skill with a method: state the task, supply the right context, show what good looks like, set constraints and fix the output format. It then goes further than tips. It gives reusable templates written out in full, a way to test prompts against real cases, a versioning habit so good prompts survive staff changes, and a clear view of what should never be pasted into a tool. It also explains how the same discipline changes between chat, document and agent settings.

WHO IT IS FOR

Team leads, analysts, operations and communications staff who use AI tools daily, and the managers and AI champions responsible for making that use consistent, safe and worth the time.

TIME TO USE

60 minutes to read; two working sessions to build your first prompt library



What you will have at the end

- A repeatable structure for writing any business request to an AI tool
- Three to five reusable templates adapted to your own recurring work
- A small test set and a scoring habit for deciding whether a prompt is good enough to share
- A shared, versioned prompt library with owners and a retirement rule
- A clear team rule on what may and may not be pasted into which tools

Contents



01	Why prompting is a business skill, not a trick	03
02	The anatomy of a good request	04
03	Context packs, examples and constraints	06
04	Output formats and reusable instruction templates	08
05	Evaluating prompts, versioning and sharing	10
06	Common failure modes and how to spot them	12
07	Chat, document and agent settings are different	13
08	What not to paste, and team rules for safe use	15
09	Rolling out a prompt library and a team habit	16
10	A first quarter that produces evidence	18
11	Terms used in this publication	19



EXECUTIVE BRIEF

Why prompting is a business skill, not a trick

01



Output quality is set mostly by what you put in: the task, the context, the examples and the limits. Teams that treat this as a skill get dependable results; teams that treat it as luck get variable ones.



A language model predicts useful text from the input it is given. It does not know your customer, your house style, your approval rules or what you meant by 'quick summary'. When the request is vague, the model fills the gaps with the most common pattern it has seen, which usually produces text that is fluent, generic and slightly wrong for your situation. People then conclude the tool is unreliable, when the real issue is that the brief was missing.

Think of the model as a capable new colleague who joined this morning. They write well and learn fast, but they have never seen your company. You would not hand them a one-line instruction and expect a finished client document. You would explain the purpose, share the relevant files, show a good past example, say what to avoid and tell them how you want the result delivered. Prompting is the same briefing, written down.



- A good prompt is a brief: purpose, audience, material, constraints, format and a definition of done
- Context beats cleverness: the right facts pasted in outperform any clever phrasing
- Reuse beats reinvention: the best prompts are saved, tested, owned and improved
- Review is part of the method: the person who sends the request owns the result
- Rules about sensitive data matter as much as rules about wording

PLAIN TEST

Hand your prompt to a capable colleague with no access to your head. If they would have to ask you three questions before starting, the model will guess the answers instead.



This playbook is written for business teams, not for researchers. It avoids folklore about magic phrases. Instead it gives you a structure you can apply to any task, templates you can adapt, a lightweight way to test whether a prompt works, and the team habits that stop good prompts from living in one person's notebook.



WORKING QUESTION

Which recurring task in your team produces the most inconsistent AI output today, and what does the model not know that a new colleague would need to be told?



ANATOMY OF A REQUEST

The anatomy of a good request

02



Six parts cover almost every business task. You will not need all six every time, but you should decide on purpose which to leave out.

Most weak prompts contain only the task, and sometimes only half of it. Strong prompts add the surrounding information that a skilled human would insist on having. The six parts below are a checklist, not a formula to recite. Order matters less than completeness, although placing the material to be worked on clearly separated from the instructions makes results more reliable and makes the prompt easier to maintain.

PART	WHAT IT SAYS	WEAK VERSUS STRONG
Role and purpose	Who the model is acting as and why the work exists	'Write an email' versus 'Draft a reply for a customer success manager to send to a frustrated customer'
Task	The single action wanted, in a verb	'Help with the report' versus 'Summarize the attached report for the executive team in 150 words'



PART	WHAT IT SAYS	WEAK VERSUS STRONG
Context	Facts, background, audience, history	None versus the customer's plan, the issue timeline and our policy on credits
Constraints	Limits on content, tone, length, sources	None versus 'Use only the attached text; no promises about dates; US spelling'
Examples	One or two samples of good output	None versus a past approved reply with a note on why it works
Output format	The shape of the answer	'Anything' versus 'A table with columns A, B, C, then three bullet risks'

Task: one verb, one deliverable

Ask for one thing at a time. A request that says 'analyze, summarize, rewrite and recommend' invites a blurred answer where none of the four jobs is done well. If you need several outputs, either ask in clearly numbered parts or chain separate requests, passing the result of one into the next. Chaining also gives you a point to review in the middle, which is where you catch an error before it spreads.

Audience and purpose change everything

The same facts are written differently for a board, a frontline team, a regulator and a customer. State who will read the output, what they already know and what you want them to do afterward. 'Write for a finance director who has two minutes and must decide whether to approve' is a stronger instruction than 'make it professional'.

Definition of done

Say what a finished answer looks like: a length, the sections it must contain, the questions it must answer, and what it should do when it cannot complete the task. The last point is often skipped and is important. Tell the model to say what is missing rather than invent it.

EXAMPLE (ILLUSTRATIVE)

Weak: Summarize this contract. Stronger: Summarize the attached supplier contract for a procurement manager. List term, renewal, price change, termination and liability clauses with section references. If a clause is not present, write 'Not found'. Do not interpret legal effect.

WORKING QUESTION

Take your last three prompts. Which of the six parts was missing from each, and what did the output get wrong as a result?

CONTEXT AND EXAMPLES

Context packs, examples and constraints

03

The facts you supply set the ceiling on quality. Build context deliberately, show one good example, and say plainly what is out of bounds.

Context is anything the model needs to know that is not in its general training: your policy, your product list, the customer thread, the style guide, last quarter's numbers. A context pack is a small, curated bundle of this material prepared in advance for a recurring task, so each person does not assemble it from memory. A pack for customer replies might hold the tone guide, the refund policy extract, a glossary of product names and three approved reply samples.

What goes in a context pack

- Purpose statement: what the pack is for and who uses it
- Current facts: policies, prices, terms, with an effective date and an owner
- Voice and style: a short guide plus one or two real samples
- Definitions: internal terms, acronyms and product names the model should use exactly
- Boundaries: topics to avoid, claims never to make, escalation triggers
- Review date: when the pack is next checked, so stale facts do not linger

Keep packs short and current. A long, unfocused pack dilutes attention and raises the chance that the model leans on an outdated paragraph. Put the most relevant material where the model will weigh it, and cut anything you would not hand to a new hire on their first day. When a document is long, supply only the sections that bear on the question and say which sections those are.

Examples: show, do not only tell

One or two good examples often teach format, tone and level of detail faster than a paragraph of description. Choose examples that are typical, not extreme, and label them as examples so the model does not copy their facts. Include a short note on why the example is good. If you give several examples that differ in style, the output will be unstable, so keep them consistent. An occasional counter-example, labelled 'do not write like this', helps when a common failure is predictable.

Constraints: be specific and positive

Constraints should be concrete and checkable. 'Keep it short' is vague; 'under 120 words' can be verified. Prefer saying what to do over what to avoid, because instructions such as 'do not mention price' keep the thing in mind. Where you need a hard limit, such as never quoting a delivery date, state it once, clearly, near the end of the instructions, and then test that it holds.

CONSTRAINT TYPE	USEFUL PHRASING	CHECK IT BY
Source limit	Use only the attached documents	Spot-check claims against sources cite section numbers
Length	Maximum 150 words; three bullet	Count
Tone	Plain, warm, no exclamation marks	Reviewer read
Scope	Do not give legal, medical or financial advice	Test with a tempting question
Uncertainty	If the text does not say, write 'Not stated'	Test with a missing fact
Terminology	Use 'client' not 'customer'	Search the output

COMMON MISTAKE

Pasting an entire folder of documents and hoping the model finds the relevant part. Select the passages that matter, label them, and tell the model what each one is.

WORKING QUESTION

For your most frequent task, which five facts would you put in a context pack, and who would be responsible for keeping them current?

OUTPUT FORMATS

Output formats and reusable instruction templates

04

Ask for a shape you can use without rework, then save the full instruction as a template so anyone on the team can run it the same way.

Format is a request like any other. If you need a table, a list of decisions, a draft email under a word limit or text a colleague can paste into a system, say so. For work that feeds another tool, ask for a fixed structure with named fields, and say what to put in a field when the information is missing. Predictable structure also makes review faster, because the reader knows where to look.

NEED	ASK FOR	WHY IT HELPS
Quick scan	Three bullets then one recommendation	Reader gets the point in seconds
Comparison	Table with named columns and rows	Differences are visible
Handoff to a person	Numbered steps with an owner per step	Actions are clear
Handoff to a system	Fixed field names, one value per field, 'Not stated' if absent	Easier to check and import
Judgment call	Options, trade-offs, then a recommendation labelled as such	Separates analysis from opinion

A template is a prompt with blanks

A reusable template holds the stable parts of the instruction and marks the variable parts in square brackets. The three templates below are written out in full so you can copy and adapt them. Replace each bracket with your material, keep the structure, and test before sharing.

Template 1: Summarize a document for a decision

You are helping a [role, for example operations manager] prepare for a decision about [decision]. Below is a document. Summarize it for [audience] in no more than [number] words.

Start with the main point in one sentence. Then list the facts that bear on the decision, each with a section or page reference. Then list anything the document does not say that the reader would need to know. Use only the document; if something is unclear, write 'Unclear' rather than guessing. Do not give a recommendation. Document: [paste text, labelled with title and date].



Template 2: Draft a reply with guardrails

You are drafting a reply for [role] to send to [recipient type] about [situation]. Use the facts and policy below as the only source of truth. Tone: [tone description]. Length: under [number] words. Acknowledge the specific issue in the first sentence, state what we will do next, and give any dates only if they appear in the facts. Do not promise outcomes, credits or timelines that are not in the policy. If the facts are insufficient to reply, list the questions to ask instead of drafting. Facts: [paste]. Policy extract: [paste]. Approved example of our style: [paste, labelled as an example].

Template 3: Extract fields into a table

Read the [document type] below and extract the following fields: [field 1], [field 2], [field 3], [field 4]. Return a table with one row per [item, for example invoice or clause] and one column per field, plus a final column named Source with the page or section where you found the value. If a value is not present, write 'Not stated'; never infer a missing value. If two places give different values, write both and mark 'Conflict'. After the table, list up to three items that need a human check and say why. Document: [paste or attach].



TEMPLATE HABIT

Every template should include three lines people forget: where the source of truth is, what to do when information is missing, and what the model must not do. Those lines prevent most confident errors.



WORKING QUESTION

Which of your team's recurring requests could become a template by next week, and who would be its owner?



TESTING AND SHARING

Evaluating prompts, versioning and sharing

05

A prompt that worked once is an anecdote. A prompt that passes a small test set, has an owner and a version number is an asset.

Model output varies from run to run and across inputs. A prompt that produced a good result on the example you tried may fail on the next three. The remedy is not elaborate. Build a small test set of real cases, run the prompt across them, score the results against a short rubric and change one thing at a time. This takes an hour for a typical template and saves repeated embarrassment.

01 Collect cases

Gather eight to fifteen real inputs, including ordinary cases, an awkward one, one with missing information and one that tempts the model to break a rule.

02 Write the rubric

Choose four or five criteria such as accuracy to source, completeness, tone, format and rule-following. Score each as pass, partial or fail.

03 Run and score

Run the prompt on every case. Score blind where possible, and have a second person score a sample to check you agree.

04 Change one thing

Edit one element of the prompt, such as a constraint or an example, then re-run all cases. Changing several things at once hides what worked.

05 Set the bar

Agree the pass mark in advance. Higher-stakes outputs need a higher bar and a mandatory human review.

CRITERION	QUESTION FOR THE REVIEWER	TYPICAL FIX
Faithfulness	Is every claim supported by the supplied material?	Add 'use only the attached'; ask for references
Completeness	Did it cover every required element?	List required elements explicitly

CRITERION	QUESTION FOR THE REVIEWER	TYPICAL FIX
Format	Does it match the requested structure exactly?	Show a formatted example
Rule-following	Did it avoid every prohibited move?	Restate the rule once, near the end; test with a tempting case
Tone	Would we send this under our name?	Add a style sample and a counter-example
Handling gaps	Did it say 'Not stated' instead of guessing?	Add an instruction for missing information

Versioning and sharing

Keep templates in one shared place, such as a team wiki or document, not in private chats. Give each a name, a version number, an owner, a purpose line, the date last tested, the test cases and a short change note. When someone improves a prompt, they update the version and record why. When a tool or model changes, re-run the test set, because behavior can shift without warning. Retire prompts that nobody uses so the library stays trustworthy.

Template record: minimum fields

- Name, version and date last tested
- Owner and a backup
- Purpose and the audience it serves
- Full prompt text with bracketed variables
- Required context pack and where it lives
- Test cases, rubric and last score
- Known limits and the review step required before use
- Change note explaining the latest edit

WORKING QUESTION

If your best prompt writer left tomorrow, which prompts would your team lose, and where would you find their latest tested versions?

FAILURE MODES

Common failure modes and how to spot them

06

The same handful of failures recur across teams. Knowing their signatures makes review faster and prompt fixes more precise.

Reviewing AI output well means knowing how it tends to go wrong. The errors are rarely random. They follow patterns that trace back to the request: missing context produces generic text, vague scope produces padding, and absent limits produce confident invention. Match the symptom to the cause and you can usually fix the prompt rather than argue with the output.

FAILURE	WHAT IT LOOKS LIKE	LIKELY CAUSE AND FIX
Invented detail	Plausible names, figures or citations that are not in your material	No source limit or no gap instruction; add 'use only' and 'Not stated'
Generic output	Correct but could be about any company	Missing context pack or example; supply specifics
Wrong audience level	Too technical or too basic	Audience not named; say who reads it and what they know
Ignored constraint	Length or prohibited topic breaches	Constraint buried or vague; make it checkable and test it
Drift in long chats	Early instructions forgotten	Restate key rules or start a fresh session with a summary
Over-agreement	Model accepts a false premise in your question	Ask it to challenge assumptions and list what could be wrong
Format breakage	Table or fields come back malformed	Show a formatted example; state the exact field names
Stale information	Uses old policy or prices	Outdated context; put effective dates in every pack

Two habits catch many errors. First, ask the model to show its basis: references to the passages or lines it used. Second, ask for a short list of things it is unsure about or could not verify. Neither guarantees correctness, and models can state a reference that does not

support the claim, so spot-check them. But both turn silent guesses into visible ones, which is where a reviewer can act.

COMMON MISTAKE

Fixing a bad output by arguing in the same chat for ten turns. Edit the original prompt, add the missing information and run it fresh. The cleaner prompt becomes the reusable one.

- Treat polished prose as untested: fluency is not evidence
- Check numbers, names, dates and quotes against the source every time
- Ask a second pair of eyes to review anything external or consequential
- Keep the original source beside the output during review
- Record recurring errors and feed them back into the template

WORKING QUESTION

Which failure from the table has your team already experienced, and which prompt change would have prevented it?

SETTINGS COMPARED

Chat, document and agent settings are different

07

The craft is the same, but the stakes, the context and the review point change as the tool gains more access and more autonomy.

Most people begin in a chat window, where you can iterate and correct in real time. As tools become connected to documents, systems and actions, the same instruction can have wider effects. The discipline of the request grows stricter as the room for correction shrinks. Match the care you take to the setting.

SETTING	HOW YOU WORK	WHAT TO TIGHTEN
Chat	Interactive; you steer turn by turn	Restate rules in long sessions; do not rely on memory of earlier turns
Document-grounded tool	Answers drawn from files or a knowledge base	Say which sources count, require citations, and ask for 'not found' when absent
Workflow with fixed steps	Prompt runs inside a form, email batch process	Fixed output format, tested on many cases, no free text where a field will do
Agent with tools	The system takes steps, reads data, may write or send	Explicit permissions, a list of allowed actions, approval before anything irreversible, logs

Document settings

When the tool reads your documents, the prompt should say which documents are authoritative, what to do when they conflict, and how to cite. Remember that text inside a document can itself contain instructions, intentionally or not. Tell the tool to treat document content as material to analyze, never as commands to follow, and be cautious about feeding it files from outside the organization.

Agent settings

An agent plans and acts in multiple steps. Instructions need a clear goal, a list of what it may and may not do, limits on spending and scope, and a stopping rule. Require a human to approve any step that sends, pays, deletes or commits. Give the agent the least access that lets it complete the task. Review its log of steps, not only its final answer, since a correct-looking result can hide a wrong path.

RULE OF THUMB

The more the system can do on its own, the more of your instruction should be about boundaries and approvals, and the less about wording.

WORKING QUESTION

Which of your current AI uses are really workflows or agents in disguise, and have their instructions been tightened accordingly?

WHAT NOT TO PASTE

What not to paste, and team rules for safe use

08

Good prompts carry real context, which tempts people to paste sensitive material. A short, clear rule set prevents most data incidents.

The safest default is to treat any tool as appropriate only for the data classes your organization has approved for it. Enterprise tools with signed terms, retention controls and no training on your inputs differ from free consumer tools, and the difference should be written down for staff. Terms vary by vendor and plan, so check current terms rather than assuming. Rules should be simple enough to remember at the moment of pasting.

DATA CLASS	EXAMPLES	TYPICAL RULE (CONFIRM WITH YOUR POLICY)
Public	Published web pages, marketing copy	Allowed in approved tools
Internal	Meeting notes, draft plans, process documents	Approved enterprise tools only
Confidential	Contracts, pricing, unreleased results, source code	Only tools approved for that class, minimum necessary text
Personal data	Names with contact details, HR records, customer histories	Avoid or anonymize; follow privacy rules and your adviser's guidance
Regulated or special	Health, financial account, credentials, legally privileged material	Do not paste unless a specific approved system and process exist
Secrets	Passwords, API keys, private keys, access tokens	Never

- Remove names, account numbers and identifiers when they are not needed for the task
- Paste the passage that matters, not the whole file
- Replace sensitive values with labelled placeholders such as [CLIENT A] and restore them afterward

- Never paste credentials, even to ask for help debugging
- Do not upload third-party material you have no right to share
- Keep a record of which tools are approved for which data classes, and publish it



Privacy law, confidentiality duties and sector rules differ by place and industry, and this playbook is not legal advice. Check current requirements with qualified advisers and turn them into a one-page team rule. Pair that rule with a no-blame route for reporting an accidental paste, because early reporting limits harm and silence does not.

EXAMPLE (ILLUSTRATIVE)

A team lead wants help rewriting a performance review. Instead of pasting the review, they describe the pattern with initials and no identifying details, ask for a structure for the conversation, and write the specifics themselves.



WORKING QUESTION

Could every person on your team name, without looking, which data classes they may paste into which tool?

ROLLING IT OUT

Rolling out a prompt library and a team habit

09



Start with a few high-use templates, test them, name owners and teach the method in short working sessions rather than long lectures.



01 Find the repeat work

List the tasks each person does weekly that involve reading, writing, summarizing or extracting. Pick the five with the most volume and the least risk.

02 Draft and test

For each, assemble a context pack, write the template and run it on ten real cases. Score and fix.

03 Publish with owners

Place templates in the shared library with owner, version and review step. Announce them with one example each.

04 Teach by doing

Run a one-hour session where people bring a real task, use the structure and compare results.

05 Review monthly

Check usage, collect failures, update versions and retire unused templates.

Measure what matters to the team. Useful signals include the time a task takes with and without the template, the share of outputs that pass review without rework, the number of corrections reviewers make per output, and the number of people using shared templates instead of private ones. Record these before you begin, since without a baseline you cannot show improvement. Treat any time saving figure as an estimate until you have timed real work.

ROLE	RESPONSIBILITY	TIME COMMITMENT (ILLUSTRATIVE)
AI champion	Runs sessions, curates the library, gathers feedback	A few hours per week at the start
Template owner	Keeps one template and its context pack current	About an hour per month each
Team lead	Sets which tasks are in scope and the review standard	Part of normal management
Security or privacy lead	Maintains the data rules and approved tools list	Periodic review

Launch readiness

- Approved tools and data rules published on one page
- Five templates tested with scores recorded
- Owners named for each template and context pack
- Review step defined for anything external or consequential
- A place to report bad outputs and accidental pastes
- Baseline timings captured before rollout


ILLUSTRATIVE SCENARIO (HYPOTHETICAL)

A regional sales support team shares a handful of prompts in a chat thread. Results vary by person and one prompt includes an outdated discount rule. The team moves the prompts to a versioned library, adds an effective-date line to the pricing context pack, tests on fifteen real requests and names an owner. Corrections fall because the rule is now maintained in one place.


WORKING QUESTION

Which five tasks would you put into the first version of your library, and who would own each?

30 / 60 / 90

A first quarter that produces evidence

10



Build a small, tested, owned library of prompts first, then widen it as the habit takes hold.

First 30 days: ground it

- List the ten most repeated reading and writing tasks and choose five
- Publish a one-page data rule listing approved tools and what may be pasted
- Capture baseline time and rework for the five tasks
- Draft context packs for the first three tasks



Days 31 to 60: test it

- Write and test templates on ten real cases each, with a shared rubric
- Name an owner and a version number for every template
- Run two working sessions where staff apply the method to live tasks
- Set the review step for external or consequential outputs



Days 61 to 90: decide

- Open the shared library to the whole team with a change log
- Compare time and rework with the baseline and report honestly
- Re-test all templates after any tool or model change
- Decide which tasks move to document-grounded or workflow settings

GLOSSARY

Terms used in this publication

11



TERM	MEANING
Prompt	The instruction and material you give an AI tool to produce an output.
Context pack	A curated bundle of facts, style guidance and samples prepared for a recurring task.
Template	A saved prompt with bracketed blanks for the parts that change each time.
Constraint	A specific, checkable limit on content, length, tone or sources.
Few-shot example	A sample of good output included in the prompt to show format and tone.
Grounding	Basing an answer on supplied material rather than the model's general memory.
Hallucination	A fluent statement that no source supports.
Rubric	A short list of criteria used to score outputs consistently.
Test set	A fixed group of real cases used to compare versions of a prompt.



TERM	MEANING
Prompt injection	Text inside a document or message that tries to give the model unintended instructions.
Agent	A system that plans and takes multiple steps, possibly using tools, toward a goal.
Data class	A category of information, such as public or confidential, that sets where it may be used.

A note on use

This publication is a public orientation resource, not legal, technical, financial or professional advice. Adapt it to your people, information, workflows and obligations, and check current tool terms, regulations and risks before acting.

TAKE THE NEXT STEP

Take the free Halden assessment at halden.company/assessments, or talk to us at halden.company/contact about turning this into a plan for your organization.