You've Got Astra. Put It to Work.

Practical GPT-6 Astra for people running a business, not spending all day chatting with AI.

Kristian | KK make IT

Free practical guide | English edition

Public working edition to try, not a fully tested final manual. Open practical tests are listed inside. The fictional Czech case retains CZK amounts.

Contents

Let's get one thing straight first 3

1. Download it. Open it. Give it its first job. 4

2. This is how I run my business. This is what you prepare for me. 7

3. An enquiry came in. Now turn it into a quote. 11

4. Have it prepare the work, not another essay. 15

5. Which Job Is Still Stuck? 20

6. Done. Now Issue the Invoice. 24

7. You're Out. Your Computer Is Working at Home. 29

8. When It Crashes, Don't Start Another Mess. 34

9. Another Customer Will Turn Up Tomorrow. 37

Appendix: what to open and what to check 40

Me and Stupid Questions: Astra's on Shift 42

Sources and edition 43

Let's get one thing straight first

AI told me it was waiting for me to approve a paragraph. Except it hadn't shown me the paragraph. So what the fuck am I supposed to approve? A literary experience delivered by telepathy? That's exactly how you can get stuck on complete bullshit with a smart tool. It feels as if it's done the work, you're looking for the result, and meanwhile you're both trying to work out who actually approved what.

And yet I use AI to prepare books, work on projects and run agents. I don't want to throw it away. I want to get the work out of it that it can actually do without surrendering my own judgement. I also don't want to spend the entire afternoon nodding along to brilliant plans, only to discover in the evening that all I've got is a brand-new to-do list. I can make one of those myself, thanks.

This guide is for someone who sells something, makes something or does work for customers. It'll take you through one fictional job, from a note to the working documents, job tracker and invoice preparation. Working from your phone gets a chapter of its own. You don't need to buy another one of our books or know how to program. You need your own head, an available app and enough time to try the process, not merely read this sentence.

One note for this English edition: the fictional case is set in Czechia, the amounts stay in CZK, and Fakturoid is the Czech example provider used later in the book. It is not universally compliant invoicing advice. You remain responsible for checking the rules and suitable tools where you do business.

We'll distinguish what we tested ourselves from instructions based on documentation. A clean-computer installation walkthrough, issuing a real invoice in an account and a complete mobile test have not been completed for this edition. You'll find the procedure and the checks for the result, not a made-up story claiming everything passed. A free book does not mean free AI operation or free accounting services. And my experience is no guarantee that the model will behave the same way for you.

Right. Open the folder. One job is plenty.

1. Download it. Open it. Give it its first job.

If you're waiting for us to explain half the history of artificial intelligence first, no. You've got work on the table. Today we'll prepare a place to put it and check that you can get something useful back out. Your first win won't be a business run by a single sentence. It'll be an ordinary file that exists, opens and matches your source material. A ridiculously small goal? Ask the person who's already spent fifteen minutes looking for a quote that was supposedly finished.

What are you actually installing?

Astra is a model, not a program you download. In this book, we work with GPT-6 Astra in a desktop environment and a local folder on Windows. You select the model inside the app. Access to your files, browser or phone is then handled by the app and its permissions, not by the word Astra on the label. Asking a question in an ordinary chat does not mean someone has just opened a folder on your drive.

Open the official installation hub [S1]. For the route used in this book, download the desktop app for Windows, run the installer and sign in with your own account. The current documentation uses the name ChatGPT desktop app; older versions and some system information may say Codex. You'll also find the Codex environment in the product menu; that is where our local tests were carried out. For the newer Work environment, the documentation recommends working with documents and spreadsheets. We are not claiming that its individual screens are identical to the ones we tested in Codex. The desktop app itself also supports other systems; the later Remote instructions specify a Windows or macOS host, not Linux [S-R2].

To reproduce the exercises precisely, select Codex, open the local project and check that GPT-6 Astra is selected below the prompt field [S2]. Model selection is different from reasoning intensity. Leave the intensity at its default to begin with. Cranking it to maximum every time you want to add up three line items is not a workflow. You can raise it for more complicated comparisons, but compare the time and quality of the result as well.

If Astra isn't available in your menu, stop and check availability for your account. Writing "you are Astra" to a different model won't turn it into Astra. You don't need to rush out and buy a more expensive subscription. First check the current options for your account and the tools you actually need. This guide does not require your own API key and gives you no reason to paste one somewhere blindly. Installation requirements change; the current official page takes precedence over where a button appeared in an old screenshot.

One folder, not a tour of your entire drive

Extract the supplied ZIP and open `exercise-pack`. Inside, you'll find the `my-business` folder, further inputs under `stages`, and templates. Next to the package, make your own copy of the `my-business` folder only, for example `Astra-exercise`. Don't work directly in the archive. Don't give AI the root of the entire package as its project: it would see confirmations, later payments and the correct answers in advance. That would be a rather convenient test, wouldn't it?

In a new app project, select that `Astra-exercise` folder. The starting point contains four files: `business-profile.md`, `work-rules.md`, `job-tracker.csv` and `jobs/Z-0001/inquiry.md`. The last file sits two subfolders deep. MD is ordinary text with markers for headings and similar formatting; you can read it in Notepad. CSV is a table stored as text. You don't need to know how to create either of them at this point.

Check the access settings too. The exercise does not need your bank, your entire desktop or your customer inboxes. In the permissions menu, use restricted access that asks before expanding it, called `Ask for approval` in the current English documentation [S3]. When the program hits a specific obstacle, deal with that obstacle. Don't allow everything in advance just to make it stop asking. Selecting a folder is not a security lock in itself, especially if you've enabled full access somewhere else.

The exercise contains no real personal or customer data. When you move to your own business, don't start with national identification numbers, passwords or medical records. An anonymised note and a price list are usually enough for a first quote. Before handing over business data, check the service terms, your account and your authority to work with that data.

An agent does not need to know everything about you to recognise the price of one hour of assembly work.

The first prompt

In a new task over the working copy, enter:

> Read business-profile.md and work-rules.md. Prepare an overview of my business in a new file named business-overview.md: what I offer, which prices you are using, and what information is missing for the first quote. Work only here. Do not overwrite existing files. At the end, give me a link to the file that was actually saved.

Let it work. If it prints a proposed plan and asks whether it should carry it out, answer specifically: "Yes, create this new file in the working copy. Do not change any other files." Once you've approved a precisely defined job, you don't need to approve every sentence it writes. The app's permissions may still require their own approval; that is not the same as a pointless conversational question.

When it's finished, open the link to the overview. Find furniture assembly, the rate of 800 CZK per hour, transport at 400 CZK and the missing information. Compare all of it with the profile. Did the model invent electrical installation? Did it mistake a six-hour limit for one visit for your availability every day? Did it add a tax regime that the source does not contain? If something is wrong, point out the exact difference and have it correct the overview specifically. An error in the overview is not a reason to change the price list.

If the link doesn't work, ask it to verify the path and open the folder in File Explorer as well. Still seeing nothing but a chat message? Then you do not yet have a saved output. In a mode that cannot write to local files, you can save the text manually. That is a usable fallback, but don't pretend it means the local-write test passed. Those are two different things.

What did we gain here?

Our original two runs produced both the overview and the quote, and we compared the outputs with the source material. That verified work over the prepared folder, not installation on somebody else's computer. On the work machine, on 12 September 2026, we found the `OpenAI.Codex` package at version `26.908.4834.0`. That describes our environment; it is not a requirement that you use only this version.

You now have a foundation that lets you tell the difference between talking about work and actually doing it. Don't connect any more apps yet. You honestly don't need seven integrations by this evening. You need the next quote to reflect how you really run your business.

2. This is how I run my business. This is what you prepare for me.

Imagine a new person comes in to help you out. You don't tell them what you sell, where the price list is, which jobs are waiting or what they mustn't promise a customer. You just point at the computer: "Help me with the business." Then you're surprised when they ask questions. Or worse, when they don't ask and confidently make something up. No, they don't need your autobiography. But without a few basic facts, they'll happily help with a business they've assembled inside their own answer.

The sample business profile in our package is already complete. Read it so you know what we actually gave the AI. Then repeat the following process in a separate folder for your own business. The exercise price list is not advice on what you should charge. And a fictional order confirmation must never pretend to be a message from your customer.

What belongs in the profile

Start with what you do and what you no longer do. A service technician may replace parts within the scope of their work while refusing work they are not authorised to carry out. A designer delivers a design but does not automatically buy advertising space. A custom manufacturer needs confirmed dimensions before production. Ordinary distinctions like these matter more to a good answer than telling the model it is the most experienced expert in the galaxy.

Add a price list with units and an effective date. Is eight hundred for an hour, an item, a visit, or one look at your calendar? Is transport charged once or for every journey? Do you price materials from your own list or from the current purchase cost? Where exactly is the reliable source? If the price varies by job, state the condition instead of inventing a falsely universal figure. Keep one current version and label the old ones as an archive, not as more equally valid price lists.

Next come capacity and the way appointments are agreed. Our sample fitter has a limit of six hours for one visit. We don't know which days are free, though. AI therefore must not promise Tuesday simply because it fits neatly into the quote. In your own business, separate your usual working hours, genuinely available dates and the date a customer has already accepted. Keep an estimate separate from a fixed price too. Those things blur together easily on the phone. You can separate them in a document.

Finally, add a few examples of your own communication. Two messages you'd genuinely send and one you can't stand are enough. Anonymise them. For the bad one, explain why: too long, needlessly servile, promises a result without the source material, sounds like a flyer advertising everything. Don't ask only for a "human tone." A formal letter that has you searching for the end by the second sentence was written by a human too.

Have it turn your notes into working material

You don't have to typeset all of this yourself. With your own anonymised notes attached, enter:

> From the attached notes, prepare a one-page profile of my business. Separate services, prices and units, capacity, how arrangements are made, and communication style. Do not fill in anything missing by estimating it. At the end, ask only the questions without which the next quote cannot be prepared. Save the profile as a new draft and leave the original source material unchanged.

Review this draft properly once. It can then support later jobs. If you discover that two of your notes give different prices, resolve the conflict at the source. You don't want whichever note the model happened to read last to win every time. When you change a price, add its effective date and say whether the change applies only to new quotes. Repricing a job already under discussion is not technical file housekeeping.

This is also where a critical question can help. "Why am I offering a date before I know when the material will arrive?" or "Why am I confirming this by phone at all when it leaves me with no record afterwards?" AI can find the contradiction and propose a better process. Whether that change fits your operation has to be judged by someone who understands it. We're not fixing the entire business in this book. We'll fix one obstacle blocking the current job.

What it can do without a performance around every step

A useful rule for everyday work is this: prepare new internal drafts directly in the designated folder; show the difference before changing an existing tracker; stop before making a commitment to a customer. Adapt those boundaries to the risk. Fixing a typo in a working draft is not the same as giving a 10,000 CZK discount. If you turn both into the same approval ritual, you'll stop reading the important one too.

The rule for the tracker is deliberately stricter in the supplied exercise so you learn to review a change. Keep it throughout the walkthrough. Don't change the test conditions halfway through and then compare the results as if nothing happened. You can set different permissions for your own routine later, but first you need to know what you're giving up control of.

Also specify what the end of the work should look like. For example: a link to the new file, what changed, what it did not do, and the one question that is genuinely blocking you. You don't need twenty bullet points celebrating the successful creation of a heading. You need to know whether the quote is waiting for an address or whether you can send it after your review. If it's

missing some unimportant wording, it can choose a reasonable phrase. If it's missing a price, it does not have the same freedom.

A rules file is not a magic amulet

In this book we use `work-rules.md` and explicitly tell the agent to read it. We don't assume that every client automatically loads any arbitrary file in a folder. The next time you start a new task, have it open the profile and rules again and show you which versions it is relying on. You don't want to hear that it remembers your business while it is reading a copy from six months ago.

Rules are not permission controls. You can write "never send anything" while needlessly keeping a mailbox connected with broad access. Written instructions are one layer; the app settings and the account's actual capabilities are another. When preparing a quote, it is easier not to grant sending access at all than to rely only on the agent correctly understanding every exception you've written.

And no, don't add another paragraph to the rulebook after every bad word. First work out what failed. An unclear source? An unavailable file? Conflicting rules? A model error despite a clear prompt? Fix the right place. Otherwise, over time, you'll build a business where almost nothing is allowed and somehow everything is still a mess.

When you start praising everything again

I watch this in my own work too. AI can respond so agreeably that you notice only the result: another idea, another plan, more work. So I give it permission to disagree and, at the same time, require it to say specifically what it disagrees with. "Be critical" by itself can easily produce nothing but a more elegant lecture.

> Do not assess my idea with praise. Find one crucial assumption that we have not yet verified and the cheapest way to verify it. If you do not see a serious contradiction, do not invent one for effect. Then continue with the agreed work. Put new ideas aside; do not use them to expand the scope.

That answer does not have to be nice. It has to be useful. And the reverse applies too: if AI shows evidence that you were wrong, there is no point beating it with prompts until it says the opposite. Did you hire an assistant, or a more expensive way to have the last word?

The profile is now the foundation of the collaboration, not another month-long project. You have what you do, the basis for your calculations and the points where the work must stop. Let's turn that into a quote.

3. An enquiry came in. Now turn it into a quote.

You spend all day doing work you know how to do. Then, in the evening, you're staring at a blank document again because you still have to tell the customer what it'll cost. It's all in your head. Three hours of work, some materials, a journey. Except two other people called in the meantime and suddenly you can't remember whether the first one wanted the table or the second one did. This is where AI should do some bloody work. You don't need a strategic opportunity for comprehensive interior development. The customer wants a table, for fuck's sake.

We'll use one practice job. Assembly of a freestanding cabinet and a table, customer A, number Z-0001. It's fictional, and so are the prices. It isn't a reference we've embellished. With a simple example, you need to be able to see where every amount came from. When you quote for design work or manufacturing a part instead of a cabinet, you replace the source material. You don't have to reinvent the whole business.

First, find the note

Continue in your `Astra-exercise` working copy from Chapter 1. Do not reopen the original `my-business` folder from the package as a project. The working copy contains `business-profile.md`, `work-rules.md`, the `job-tracker.csv` job tracker and the `jobs/Z-0001` folder. Open `inquiry.md` in that last folder. Files ending in MD are text source materials; you can read them in an ordinary text editor. There is no programming hiding behind them.

The note says the customer has the furniture at home, still in its packaging. The exact address, access to the flat and the date are missing. For the exercise, the business owner wrote down their own estimate: three hours of work, materials costing 1,200 CZK and one local journey. The profile gives a

rate of 800 CZK per hour and 400 CZK per journey. Astra did not decide on those three hours by peering through the phone into somebody else's living room. They were supplied by the person who prices the work.

Record your own enquiries this way too. Separate what the customer actually said from what you estimated. If you're unsure, write that down. "About three hours, verify against the instructions" is more useful than a confident three with no note. The customer may not remember exactly what was said. Neither may you. That is precisely why we're making a record.

Give Astra a specific job

In the same Codex task where Astra has already read the profile and rules for this working folder, enter:

> From enquiry Z-0001, prepare a draft quote in jobs/Z-0001/quote-v1.md. Use my price list and my work estimate from the note. Itemise the amounts. Leave anything not agreed as a question. Save a short reply to the customer separately in jobs/Z-0001/customer-reply-v1.md; no more than 120 words, containing only the text for the customer. Save internal checks, missing source material and the proposed tracker change separately in jobs/Z-0001/internal-notes-v1.md. Do not send anything. For job-tracker.csv, prepare only a proposed change to row Z-0001, setting the status to quote_draft. Do not overwrite the CSV yet; writing the change must wait for my separate approval of the specific proposal.

This prompt has been tightened for the current package. Unlike the two original tests, it explicitly separates the customer-facing text and prohibits writing to the tracker before approval. It applies together with the profile and rules in your folder. The same sentence in an empty chat would not be the same task. The filename tells the agent where the result must end up. Do not approve the tracker change at this point.

When it finishes, open `jobs/Z-0001/quote-v1.md`. Reading a reply that says "I saved it" is not enough. Click the link and read the contents. If the file does not open, have Astra verify that it exists and return a working link. Until you have a readable file, you have not accepted the quote.

What exactly are you accepting?

Start with the line items. Our exercise must produce:

- Labour: 3 hours at 800 CZK, making 2,400 CZK.

- Materials based on your list: 1 set at 1,200 CZK.
- Local transport: 1 journey at 400 CZK.

The total is 4,000 CZK. No discount you never promised. No splitting the materials into twenty parts with invented prices. The note contains one set of materials, so the draft keeps it as one set for now. We did not include a tax regime in the sample profile. The result is therefore not a quote ready to send in real life. First add the supplier's real details and clarify the price, including how it should be described for tax purposes.

Now check the scope. Are both the cabinet and the table there? One journey? Your three-hour estimate? Is the date still open? Six hours available for one sample visit does not tell you that Friday is free. Check the questions too. The customer supplies the address. You verify the suitability of the materials and the labour estimate against the source material. Don't dump every unknown on the customer just because AI listed them all under one heading.

Both of our runs got the total right, saved the file and preserved the missing date. The tracker remained unchanged, which we verified by comparing the original files. That is a specific result from two tests while preparing the book. It is not proof that Astra will never screw up one of your quotes. This practice job was not carried out for any customer.

This is where you need to touch the result

Both original drafts included internal notes alongside the quote. Notes about sources, missing business details or record-keeping. That is why the new version of the rules separates them in advance. Even so, read the result: the customer does not need to know the name of a column in your table. If internal notes get mixed into the message again, separate the customer section into its own version and keep your own questions with the job. Preserve the original draft so you can identify the change later.

The short reply should help the customer move forward. Here, for example, you need the address, access details for the flat, the furniture models or instructions, and their preferred date. You do not need six sentences about how much you appreciate the opportunity to contribute to their dream home. For fuck's sake, you're a fitter, not a wedding officiant. You can still address the customer politely and write with respect. The way I write this book is not the mandatory tone for your business.

You don't have to rewrite the whole thing by hand. To separate the two parts, I suggest following up with this prompt:

> Preserve the original quote-v1.md. In customer-reply-v1.md in the same folder, put only a draft message to the customer, no more than 120 words. Use a polite, professional form of address. Preserve the price, scope and unanswered questions, and do not promise anything more. My internal checks and tracker notes do not belong in the message. Keep the information still needed to complete the quote separate for me. Do not send anything.

This is a corrective step for when the texts were not separated properly the first time. It was not part of the two historical tests and does not mean you need to generate another message every time. If a correct separate draft already exists, have it compared, not needlessly overwritten. Open the message next to the quote and compare the amounts, scope and who each question is meant for.

Astra also flagged the date in both runs: the note from the practice call was dated one day later than the actual test day. It correctly pointed out the difference and did not overwrite the source. In our exercise, the reason is the fictional calendar. In your job, it might be a typo or an old document. Resolve that sort of question once at the source instead of banning AI from bothering you.

A draft is not an order

One more detail from the test. One run proposed the status `quote_draft`; the other proposed `draft_quote`. A person understands both. A filter looking for the exact name may not. The prompt did not define a standard value. That is something we need to fix, rather than pretending the machine broke an agreement it was never given.

The current package therefore specifies exactly `quote_draft` in the rules. Before approving the tracker change, compare the proposed value and the affected row. Only the draft status and next step may change. The field for the approved amount stays empty, as does the date. We can add up 4,000 CZK. The customer has not approved it yet.

Once you've read the specific change, you can approve saving it. Then reopen the tracker and compare the row with what you approved. Remember that you are still tracking an enquiry with a draft quote. Agreeing

to change the table did not deliver an email to the customer, and their silence did not reserve Tuesday for you.

You now have what you need to finish the quote and you know what to ask about. Next time, you won't be fishing a price out of your memory between three calls. You'll open the job. And once the scope really has been confirmed, you'll have the work prepared. That table won't assemble itself.

4. Have it prepare the work, not another essay.

"Could we add one more shelf?"

We could. We could also make a second visit, buy different materials and do work nobody confirmed. So far, the customer has asked a question. You haven't said what it'll cost or when you'll do it. If the shelf moves itself from the chat into the agreed scope along the way, you've just had AI create unpaid work for you.

We're continuing with the synthetic job Z-0001. Assembly of the cabinet and table has been confirmed at 4,000 CZK. The shelf at the proposed 600 CZK has not been confirmed. Astra should prepare the work so that, in the morning, you know what to bring and what you still need to find out. It must not create the impression that everything has been agreed just because a complete-looking document is pretty.

First, get straight what is actually valid

Add `confirmation.md` and `shelf-request.md` from `stages/03-preparation/jobs/Z-0001` to the working copy. They belong to the existing Z-0001. The second message is not a second job.

The confirmation establishes the original scope, the amount of 4,000 CZK and the fictional appointment at 09:00 on 15 September 2026. The exact location and the manufacturer's source material are still missing. The later message establishes only the request for a shelf and your proposed price of 600 CZK. Not approval. Not a promise to do it on the same date. Not completed work.

Compare the confirmation with the saved quote. If the quote contained two cabinets and the confirmation contained one, the newer file would not bless the mistake. You have a contradiction that a person must resolve. A file date is not approval either. The exercise also uses a fictional calendar, so do not reset your computer because of it or present its dates as real events.

Ask for a result you can open in the morning

> The practice date is 9 September 2026. Read confirmation.md, shelf-request.md and the draft quote for Z-0001. Prepare job-sheet.md with the confirmed scope, date, materials, source material and checks before departure. Save the new request separately in changes.md with its source and actual status. Do not make anything up about the location, assembly method or completed work. Show a proposed change to job-tracker.csv, but do not overwrite the file yet. Preserve existing outputs and compare their contents first.

The output should be a job sheet, not an essay on the importance of preparation. Open it and imagine you're standing by the vehicle. Can you immediately see Z-0001, the cabinet, the table, the appointment and the materials? Can you see what is missing? If the document does not tell you where to go, your preparation is not finished. All you have is nine o'clock confirmed and, surprisingly, no ability to teleport.

Before accepting it, check that:

- the original scope is the cabinet and table, not the shelf;
- the approved amount is 4,000 CZK, not 4,600 CZK;
- the appointment is confirmed, while the location and some source material are missing;
- the type of joint or assembly method has not been filled in by estimation;
- the work is not marked as completed or handed over.

Feel free to shorten the job sheet to one page. Length is not performance. Performance means preparing the right work from it and not forgetting the open issue.

Don't approve a file you haven't seen

The assistant may announce that it has prepared the job sheet and immediately ask for approval of the next step. But what exactly are you approving? The filename? Its existence? Or contents nobody has shown you?

Always have it show you the path, the exact contents and the proposed tracker change. Then open the file yourself. A `done` message is not acceptance. We dealt with the same distinction for the quote: prepared text, customer-approved text and sent text are three different results. Here we add the physical work, which the document must not pretend has happened.

If the job sheet is wrong, don't correct it with "make it better." Name the specific mistake and the required state:

> In job-sheet.md, the shelf is listed among the confirmed items. Move it only to changes.md as ZM-001 awaiting approval. Restore the approved amount to 4,000 CZK. Do not change the other parts of the job sheet, preserve the original version, and show me the exact difference before saving.

That is a useful correction. It identifies the file, the mistake, the correct meaning and the boundary of the change. Astra does not need to reinvent the entire job, and you do not need to review another five pages you never asked it to change.

The shelf has its own status

`changes.md` should contain ZM-001, one shelf, the proposed 600 CZK, a reference to the source and the status `awaiting approval`. We do not know its effect on the date. The change must not disappear, but it must not be added to the job either.

Separate the internal and customer-facing voices here. Internally, you need the price, scheduling risk, source and next step. The customer does not need to read your notes such as "source material missing" or instructions for Astra. They need one normal question.

> Prepare two separate outputs. In internal-note.md, record ZM-001, the proposed 600 CZK, the missing approval and the unverified effect on the date. In customer-reply-draft.md, write a short message asking the customer to confirm the price and explaining that I still need to verify the date for the additional item. Do not send anything and do not change the original job.

When the actual confirmation arrives, save it as a new source. Only then do the scope, job sheet and possibly the price change. "Could we?" is not an order. You can make the shelf for free, decline it or move it to a later date. Just make it your decision, not a side effect of the model's eagerness to finish everything.

A designer needs six genuinely different drafts

We tested the same principle on a synthetic design job. The order was for six posts, with two rounds of comments agreed and a fictional price of 3,600 CZK. A seventh post was waiting as a change request. The outputs were meant to be copy and descriptions of the visuals, not publication.

The first set met the formal requirements. There were six files, with the correct topics and a polite, professional form of address. But nearly every text began with a question and ended with a similar call to action. Internal reservations also leaked into the customer-facing copy. So a complete output still wasn't a good output.

> Read the six drafts in drafts-v1 and the source file designer.md. Prepare the complete set in drafts-v2. Preserve the topics, but develop them in different ways: with a brief situation, an observation, dialogue or a specific question. Do not begin all six with a question or repeat the same call to action. Do not invent customer experiences, repair outcomes or new services. Preserve a polite, professional form of address. Put internal limitations separately below the copy, not inside the customer-facing message. Do not overwrite V1, do not create the seventh post, and do not publish anything.

In two local runs, the second set passed as a usable first working version. That does not mean customers confirmed its sales effectiveness. It means something more sober: specific criticism corrected a specific weakness. "Write it less like AI" would not have produced such a precise result.

When accepting the result, don't just count the files. Read them side by side. Six pieces of copy can have six titles and still be the same sentence dressed in six different outfits. For a customer-facing output, check meaning, rhythm and brand voice. For an internal note, check constraints and open questions. Mix them together and neither will work properly.

An internal correction to your own draft is not a customer round of comments either. Version V2 must be preserved, and the first actual

customer comments get another version. Otherwise, a month from now, you won't remember what the customer corrected and what you broke yourself.

In manufacturing, the newest file is not automatically the right one

In the synthetic manufacturing case, twenty organisers in black PETG had been confirmed according to V1, with a 20 mm opening. V2 arrived later with a 24 mm opening, but without confirmation that it replaced the ordered version. Astra correctly prepared the production sheet for V1 and left V2 as a pending change.

> From the source file manufacturer.md, prepare production-sheet.md for twenty units according to the confirmed V1 and a separate changes.md for V2. State which inputs are missing before costing and production. Do not estimate print time or consumption without a result from the relevant tool. Do not overwrite the approved version or present the sheet as permission to start production.

"Use the newest file" is a pretty stupid instruction in manufacturing. Newer can mean corrected. It can also mean unapproved. AI should trace the status and source, not guess which meaning you happened to want today.

At the end of this chapter, you do not have a completed installation, campaign or production run. You have prepared work, separated change requests and files that can be accepted. That is less impressive than claiming "Astra solved the job." But you can actually work from it.

5. Which Job Is Still Stuck?

The customer is asking whether you've forgotten about them. You haven't. You just don't know whether you sent the quote or merely finished writing it that evening and then went to bed. You open your email, your messages, and three files with `final` in the name. Meanwhile, the customer gets nothing but the sound of you hammering away at the keyboard.

You don't need to build your own CRM for three jobs. You need one job tracker where every job has exactly one current row, the source of the latest change, and a next step. Astra prepares a proposed update from the source material. You check whether the same event has been added twice.

Three Jobs, Three Different Problems

Add the contents of `stages/04-overview` to your working copy. The scenario date is the morning of 16 September 2026. All events are synthetic.

In this scenario, the original cabinet and table assembly for Z-0001 has been completed and handed over to the customer for CZK 4,000. The shelf for CZK 600 was neither approved nor delivered. No invoice has been created, and there is no evidence of payment.

For Z-0002, a quote for CZK 2,000 was sent. The customer's decision and the date are both missing. The quoted amount therefore does not belong in the approved-price field.

Z-0003 has been delivered, has the synthetic document `CVICNY-Z0003-01` for CZK 6,000, and has one documented practice payment of the full amount. You don't issue another document here. You check the links between the records and save the source material.

Add up every number and you get CZK 12,000. And what exactly is that? Money received, uninvoiced work, and a quote with no response, all thrown into one bag. Lovely for a colorful chart. Completely useless for deciding how much money is in your account.

One Tracker, Not Three Versions of the Truth

The main file is `job-tracker.csv`. Work status, document creation, sending, and payment are separate pieces of information. `today-overview.md` is only a readable view of the confirmed tracker, not somewhere to fix reality on the side.

> It is the practice morning of 16 September 2026. Read the new documented events and prepare a proposed update to the single job-tracker.csv table. Keep work, document, sending, and payment status separate for each job. Use the exact states from work-rules.md. Include the source of the latest change and the next step. Preserve the existing records, check the IDs and events already processed, and show the changes. Stop there and wait for separate confirmation of the specific rows. Do not change the CSV yet, and do not create today-overview.md yet.

Before approving anything, you need to see this meaning:

- Z-0001: delivered, CZK 4,000, document not issued, nothing sent, CZK 0 paid; the next step is to prepare the invoice details;
- Z-0002: quote sent, acceptance unknown, approved amount blank; the next step is to find out the customer's decision;
- Z-0003: delivered, practice document issued and sent, CZK 6,000 paid; the next step is review and archiving.

A blank approved amount for Z-0002 doesn't mean the work is free. It means the customer hasn't confirmed the price yet. An unknown value isn't zero, and silence isn't consent. These are exactly the little details that dashboards love turning into an optimistic fairy tale.

Status Is Not the Next Step

`Delivered` describes what happened to the work. `Prepare invoice details` says what you need to do next. If you cram both into one column, a week later you'll be filtering values such as `done waiting for invoice will probably pay`. That isn't a status. It's the diary of someone in a mild panic.

Keep the meaning of the fields simple. Choose the work status from a short vocabulary that you use consistently everywhere. Write the next step as a specific action. The source shows why the status changed. Standardize the exact state names in your own pack before using automated filters; in our two original Czech-language trials, the names differed for the same meaning

of completed and delivered work. It wasn't a factual error, but to a filter, `completed_and_accepted` and `completed_handed_over` are still two different values.

The current practice pack already contains a standardized vocabulary in `work-rules.md`: for example, `completed_handed_over` for delivered work. In your existing tracker, though, don't let Astra casually replace every value across the entire history. Show the mapping first and check what the downstream views use. Otherwise, you'll remove one inconsistency and create three new ones.

Approval Belongs to a Specific Change

Open the source for every proposed change. For Z-0001, that's the handover; for Z-0002, the quote; and for Z-0003, the document summary with its payment. Only then should you approve the specific rows.

Before writing, Astra needs to reload the current CSV, compare it with the version used to prepare the proposal, and keep a backup. If you've changed something manually in the meantime, the old approval doesn't cover overwriting the new work. Open the CSV again after the write. It must contain three different IDs, and Z-0003 must have one payment of CZK 6,000 linked to the correct document.

Your confirmation could read: "I approve exactly the displayed rows Z-0001 through Z-0003. If the source CSV hasn't changed, back it up and write these changes. Add the processed events to processed-events.md. Then create today-overview.md from the saved tracker and show me both files." Don't send that automatically just because the next paragraph of the book carries on.

The status doesn't change in the morning overview. It changes in the tracker based on a source, and only then is a new overview generated from it. Otherwise, the CSV will contain a pending quote, the Markdown will contain a finished job, and your phone will tell a third version of the story. A system like that doesn't need Astra. It needs tidying up.

The Overview Must End in Work

At the top, `today-overview.md` must state the scenario date and the source version of the tracker. Below that, the status and next step are enough. For our morning: prepare the invoice details for Z-0001, find out the decision on Z-0002, and finish the review of Z-0003.

You can change the order based on information that isn't in the folder. If the second customer is leaving in half an hour, tell Astra or just call them. A folder on disk isn't telepathy.

Then present the same three events a second time and ask only for a check. The model must open both the tracker and the event records. It isn't enough that the conversation remembers something.

In two original Czech-language local trials, reloading the events did not change any rows or amounts. There were still three jobs and one payment of CZK 6,000. We tested this scenario, not the infallibility of every future tracker. We did not test concurrent manual edits or the same ID with different content. The translated English prompts have not had the same live model trial.

The same ID with different content is precisely what must not be silently discarded as a duplicate. If E-006 says CZK 6,000 once and CZK 8,000 the next time, you don't have an event that's "already processed." You have a conflict. Preserve both versions, add nothing to the totals, and send the decision back to a person with the sources side by side. Automatic deduplication is useful right up until it starts automatically hiding the problem.

Now you know which job is stuck and why. Close the overview and take the first concrete next step. Your business doesn't make money when you sit there happily staring at a perfectly sorted list.

6. Done. Now Issue the Invoice.

The work has been delivered. Great. That doesn't make the money arrive by itself, and a worksheet doesn't turn into an invoice.

For Z-0001, we have synthetic confirmation that the assembly of the cabinet and table was completed and accepted for CZK 4,000. The shelf for CZK 600 was neither approved nor delivered. It doesn't belong on the invoice. You can't add it just because the customer mentioned it once in a message. That wouldn't be automation. It would be confidently billing for something that never happened.

This specific route is based on Fakturoid's official documentation [S-F1 through S-F5]. We haven't yet completed a practical issuance test in a test account. What follows is therefore a specific guide, not a claim that the test was performed successfully. This remains a Czech CZK case using the Czech example provider Fakturoid, not US tax guidance or universally compliant invoicing advice.

First You Create the Details, Not the Invoice

Add `stages/05-invoicing/invoicing/Z-0001/invoice-input.md` to your working copy. The file itself tells you what's missing: both parties' identification, the tax treatment, payment details, the due date, and a check of the specific invoicing environment. These aren't decorative boxes for Astra to fill in creatively.

Ask it to prepare the internal invoice details:

> For the synthetic Z-0001, prepare invoicing/Z-0001/ready-to-issue.md. Compare confirmation.md, handover.md, shelf-request.md, and invoice-input.md. Carry over only the delivered items: three hours of work at CZK 800, material for CZK 1,200, and one trip for CZK 400. Recalculate the total. List the CZK 600 shelf separately as an unapproved, undelivered change outside the invoice. List the missing details, the environment, and the check for whether a document already exists for Z-0001. Prepare a summary for human confirmation. Do not issue or send anything, and do not mark anything as paid.

In `ready-to-issue.md`, you want to see CZK 4,000 and a clear list of the missing details. Not an invoice number the model made up. Not a fake company registration number. Not the sentence "document created." Until the invoicing application actually creates the document and shows you its identity, all you have are invoice details.

Before the next step, open Fakturoid and search by the customer, amount, and your own job reference to see whether the document has already been created. A file on your disk doesn't know the current state of the account. If an earlier attempt stopped after the click, don't make a second one. Search first. Two identical invoices aren't proof of diligence.

The Specific Route in Fakturoid

For this chapter, we're using the manual web interface. No API and no connecting the account to AI. Astra prepares and checks the details. You work in an existing account whose invoicing settings you know.

In Fakturoid, open `Faktury` (`Invoices`) and click `Nová faktura` (`New invoice`). According to the help documentation, you can open the same screen with the yellow + button in the main menu. The Czech labels are the click targets verified by the source; the English text in parentheses is only a gloss. Then:

- Select the real customer and verify their details. We don't have the required details for our synthetic customer, so you stop the exercise here.
- Check your own invoicing details and tax settings directly in the account. Astra must not decide whether you're registered for VAT.
- Select the payment method and set the due date according to your actual agreement.
- Add the delivered items, quantities, and prices. For Z-0001, their practice total would be CZK 4,000. The CZK 600 shelf stays out.
- Under `Další možnosti` (`More options`), you can add Z-0001 to the `Číslo objednávky` (`Order number`) field. It's a useful link, not a substitute for checking the customer and amount.
- Compare the form with `ready-to-issue.md`, the confirmation, and the handover.

The Final Check Before the Yellow Button

First, compare the summary on screen directly with the source details. If you want Astra's help, give it readable, anonymized form data or a screenshot; simply having a tab open in your browser doesn't give it access. You need to see the environment, customer, items, quantities, prices, total, tax settings, payment method, due date, and order number. The last line must say exactly what the subsequent confirmation allows and what it doesn't.

> From the form and ready-to-issue.md, prepare only a review summary. Include the account or environment name, customer, every item, quantities, unit prices, total, tax settings, payment method, due date, and order number. Highlight every discrepancy. End with: "After this review, I must make a separate human decision before I manually create one invoice for this customer, amount and account. This review does not permit Astra to create or send anything or to record a payment." Do not take any external action.

This isn't pointless approval theater. With a text draft, you can afford a quick correction. With an accounting document, you need to see who and what you're invoicing before the change happens. At the same time, you aren't reviewing ten intermediate steps. You're reviewing one precisely described action with a clear consequence. The actual click still requires a separate human decision.

If the summary doesn't match, go back to the form. Don't tell Astra to "make it add up somehow." The difference between CZK 4,000 and CZK 4,600 isn't cosmetic, and an unknown tax treatment isn't room for creativity.

Only the yellow `Vytvořit fakturu` (`Create invoice`) button creates the invoice. Until then, you're preparing a form or draft. The click belongs to a specific human decision about a specific customer, amount, and environment. Consent along the lines of "yeah, carry on with the chapter" really isn't consent to issue an accounting document.

After creation, open the invoice detail and record its actual number, amount, date, and link to Z-0001. The `invoice_status` field can change from `not_issued` to `issued`. The work status doesn't change with it. Sending still remains `not_sent`, and payment remains CZK 0. One step, one change.

Don't write only the number that appeared on the screen into the tracker. Save the source of the check and the time when you opened the detail as

well. If the application refuses the action or the result isn't clear, the status remains unverified. An error message isn't an issued invoice, but it also isn't a reason to click blindly a second time.

After every step, close the detail and reopen it from the overview. That verifies the saved state in the application, not just the last screen you remember seeing.

Sending Is Another Button

According to Fakturoid's help documentation, you send an invoice with the `Poslat` (`Send`) button directly from its detail. From the invoice list, there is another route through the three-dot menu and `Poslat emailem` (`Send by email`). Neither one is triggered merely by creating the invoice.

Before sending, check the recipient, any copied addresses, and the message text. The internal review from `ready-to-issue.md` doesn't belong in the email. The customer doesn't need to read that the model was looking for a missing company registration number or that the shelf is waiting in ZM-001. They need to know which work they actually received the invoice for and when it's due.

> From the issued invoice detail and the approved handover, prepare a short customer email in email-draft.md. State the cabinet and table assembly, invoice number, and due date exactly as shown on the document. Do not add internal notes, the unapproved shelf, or any claim that it has been paid. Do not send anything.

Read the text in the preview, verify the address, and only then make a separate decision about `Poslat` (`Send`). After it has actually been sent, check the status in the invoice detail and only then change `invoice_sending` to `sent`. An issued invoice sitting in an email draft isn't a sent invoice.

Paid Means You Have Evidence of Payment

The third step comes only when you have documented payment. Not when the customer writes, "I've sent it." Not when two days have passed. When you have a real bank or cash record and can match it to the correct invoice.

In the invoice detail, click `Zaznamenat platbu` (`Record payment`). According to the help documentation, the same option is available from the three-dot menu in the overview. Enter the actual date and the amount actually

received. The form distinguishes full and partial payment. If you enter less than the full amount, Fakturoid automatically turns off the full-payment toggle. Don't turn it back on just to make the green status look tidy.

The pack contains a separate offline exercise: `test-document.md` for CZK 4,000 and `test-payment.md` for CZK 2,000. The first file isn't an invoice, and the second isn't a bank statement. They belong only in a separate offline copy for practicing the record-keeping logic. The correct result is CZK 2,000 paid, CZK 2,000 remaining, sending unknown, and no due date supplied. Don't enter anything in Fakturoid based on them.

> Only in the separate offline copy, compare test-document.md and test-payment.md. Save the result in invoicing/Z-0001/test-payment-status.md: document ID, payment ID, sources E-007 and E-008, amount, currency, paid, and remaining. Do not claim that the document was sent or is overdue. If the file already exists, read it first and verify the same single PL-CV-001; do not add it a second time. If there is a conflict, do not overwrite anything. Do not change job-tracker.csv, the main working copy, or any external accounts.

This lets you practice the logic without creating fake accounting records. In real work, though, you change the `paid` status only on the basis of actual evidence and the detail of the specific invoice.

Four States That Must Not Be Blurred Together

Approved work tells you what you're allowed to invoice. Issuance tells you that a document was created in the application. Sending tells you that a customer message went out. Payment tells you how much money has documented evidence linking it to the document.

It sounds obvious. Except that as soon as you blur it all into the single word `done`, you can have perfectly completed work, no invoice, and no money. Then you'll be explaining that the system did mark everything green, after all.

Astra is there to prepare the invoice details, recalculate the items, find discrepancies, and draft the customer message. Fakturoid is there to create and record the real document. You decide whether to issue it, send it, and allocate a payment. Three different roles. And now you finally know who did which part.

7. You're Out. Your Computer Is Working at Home.

Your phone isn't a second computer that secretly materializes at your desk back home. It's a remote control. You send the task from your phone, but the work is done by the connected host: the computer where the project, files, sign-ins, permissions, and local tools live.

That distinction matters. Otherwise, you'll end up with a nice answer on the screen and assume a file changed at home. It didn't. An answer isn't a file, and the words "all done" certainly aren't proof.

Requirements First, Pairing Second

The current official guides refer to the **ChatGPT desktop app**. Depending on the kind of work, you can choose ChatGPT Work or Codex in it. An older installation may still use a different package name or may already include Codex, but this guide shouldn't rest on a historical version number. It rests on what you can actually see today.

Before you start looking for a QR code, check four requirements:

- Your account and workspace give you access to Codex.
- Your phone has the current ChatGPT app for iOS or Android, and you can see Remote in it.
- The host is a supported Mac or Windows computer running the current ChatGPT desktop app. The computer is awake, online, and signed in to the same account and workspace.
- The workspace administrator hasn't disabled Remote Control.

Remote is rolling out gradually. If you can't see it on your phone, the answer isn't to start pressing random buttons, creating new accounts, or installing something from a dubious guide. Update the apps first, then check feature availability and the workspace rules. If the requirements aren't met, you can't pair the devices right now. You've saved yourself from hunting for a button you don't even have.

Remote isn't installed as a separate model or as some random add-on from the internet. You need the supported feature in the current mobile and desktop apps. If you're missing the ChatGPT desktop app itself, return to the installation process in the first chapter and use the official download for your system. We aren't bolting anything else on here, and we definitely aren't setting up a paid API for it.

Pair a Device You Own

Setup starts on the computer, not in the CLI and not in an editor extension. In the ChatGPT desktop app, open **Settings > Connections > Control this Mac or PC** and choose **Set up** or **Add**. Allow remote access and complete whatever verification the app requires.

Only this process displays the QR code. Scan it with your own phone, let it open ChatGPT, confirm the same account and workspace, and complete any MFA, SSO, or passkey verification. Pair only a device you trust. Under **Settings > Connections**, you can then see and manage connected devices.

The phone doesn't replace the host. If the computer goes to sleep, loses its network connection, or the app closes, remote access stops. Tasks that use Computer Use on Windows also require the session to remain available and unlocked; Computer Use works in the foreground. You don't automatically need Computer Use for ordinary file work, though. Don't unlock half your computer for this chapter, and certainly don't disable security across the board.

Before you leave, run the boring test that may save you a trip home later. Keep the computer connected to power, leave the app running, and use the available setting to prevent sleep during remote access. Then, while you're still at the desk, open Remote on your phone and check that you can see the correct host and project. The test task can simply read the name of a practice file. It doesn't need to change anything. If the host disappears while you're still next to the keyboard, it won't miraculously sort itself out when you're outside.

From Your Phone to the Right Job

In the mobile app, open Remote and select the correct computer and project. You can create a new task or continue an existing one. For our exercise, continue with the original working copy where you added `stages/06-mobile`. Don't use the copy with the substitute simulated invoice from the previous chapter.

Before you send the task, write down the host, working folder, job ID, and latest known version. Sound bureaucratic? Maybe. But guessing on your phone whether you're editing Z-0002 on your home computer or an old copy somewhere else is even more stupid.

Job Z-0002 has a new synthetic event. The customer originally received quote v1 for CZK 2,000. Now they want four cabinets instead of two. The practice estimate is four hours of work at CZK 800 plus one local trip for CZK 400. The customer supplies the material, and no date has been agreed. The v2 draft therefore comes to CZK 3,600.

> I'm working on host [name] in working folder [path]. For job Z-0002, read clarification.md and preserve quote-v1.md unchanged. Prepare quote-v2.md and a short summary of the changes from v1. Calculate four hours of work at CZK 800 and one trip at CZK 400, for a total of CZK 3,600. The customer supplies the material, and no date has been agreed. V2 is only a draft for my review. Do not send anything, and don't change the fact that the customer received only v1. At the end, show a link to the saved file.

Here's the crucial difference: the new price isn't a correction to history. V1 for CZK 2,000 really does remain the version the fictional customer received. V2 for CZK 3,600 is a draft based on the new scope. If you silently overwrote v1, the records would later claim the customer received something they never got. AI wouldn't have created a quote. It would have created a dispute for you.

Open the Result. Actually Open It.

Wait for the task to finish, then open the link to `quote-v2.md` on your phone. Don't read only the final summary. Check the file for:

- four cabinets and four hours of work;
- CZK 3,200 for the work, CZK 400 for the trip, and CZK 3,600 in total;
- material supplied by the customer;

- the date still left open;
- v2 identified as a draft, with nothing sent;
- v1 for CZK 2,000 preserved as the sent version.

Find one specific sentence and have it changed. For example, instead of the vague "the date will be confirmed," enter:

> In quote-v2.md, replace the sentence about the date with the exact sentence "The installation date hasn't been agreed yet." Show the proposed change first. After I confirm it, edit only this v2 draft. Don't change v1, and don't send anything.

Then open the file again. Yes, again. Checking before a change and checking after a change are two different things.

You still need to feed the change back into the central job tracker. Otherwise, you'll have a new quote on your phone while the table keeps telling you to chase an answer to the old one. Once again, the task is only to prepare a proposal:

> Based on E-009, prepare a change to only the Z-0002 row in job-tracker.csv. The latest version, v2, is quote_draft; the approved amount and date remain blank. The latest event will be E-009, sourced from jobs/Z-0002/clarification.md. Next step: review and finish v2 for CZK 3,600. Preserve in the note that v1 for CZK 2,000 was sent and v2 was not. Do not change invoice_id, invoice_status, invoice_sending or paid_czk. Also propose an E-009 entry for processed-events.md. Show the exact changes and wait; don't write any of them yet.

After a separate confirmation, have it compare the source tracker again and only then write the approved changes and event. Open both files. There will still be three jobs, E-009 will be processed once, and acceptance of the quote will still be unknown. You don't erase the old, sent v1 by marking the latest working version as a draft. The original quote, sources, and event record preserve the history, not one magic word in a table.

When the Connection Drops

Don't create a second task just because your phone briefly disconnected. First find out whether the host is awake, online, and running the app. Then open the original task and look at its latest state. Finally, verify the files: does v2 exist, does it contain the expected content, and is v1 still there? Is the tracker change still a proposal, or was it approved and saved with E-009 recorded?

Only then should you continue. Sending the whole prompt again could create another version, overwrite a draft in progress, or repeat a later action you don't yet know about. In the next chapter, we'll turn this into a short handoff so you don't have to rely on memory after a failure.

We did not perform this mobile walkthrough in practice while preparing the book; the steps are verified against the current official documentation [S-R1, S-R2], not by a practical test on a phone.

8. When It Crashes, Don't Start Another Mess.

A failure isn't an instruction to "do everything again."

That applies to a lost mobile connection, a closed app, an interrupted response, and handing work over to another context. You don't know how far the action got? Fine. Then the first step isn't repetition. The first step is finding out the state.

With a text draft, a duplicate attempt may mean two similar files. Annoying, but you'll survive. With an order, invoice, payment, or customer message, it can create a second commitment. At that point, "I thought it had crashed" isn't going to interest anyone much.

The Response Was Interrupted. The Action Might Not Have Been.

When the assistant's final sentence disappears, that doesn't automatically mean nothing happened. Likewise, a button spinning for a while doesn't prove success. Look for the state wherever the result should have evidence:

- for a quote, in the working folder and version history;
- for the tracker, in the single `job-tracker.csv` and the source event;
- for an invoice, in the specific invoicing application by its number or ID;
- for a sent message, in the sending history;
- for a payment, in the banking or accounting source, not in a confident sentence from AI.

Evidence first. Next step second.

It helps when every event has its own identity. In our exercise, the new clarification is E-009 and relates to Z-0002. In real work, that could be an order ID, document number, timestamp of a received message, or unambiguous filename. When you load the same event again, the system should recognize it as the same event, not as a new instruction. Without an

identity, you're comparing two sentences by feel. With one, you can say: this one has already been processed; this one hasn't.

A Handoff That Isn't a Novel

A new context doesn't need your company's entire life story or thirty screens of an old conversation. It needs enough information to avoid repeating work and guessing the wrong state.

Write down:

- which task, host, and working folder you're dealing with;
- the last step you know is confirmed;
- the file or record that really exists;
- what a person actually checked;
- which action was interrupted;
- what you still don't know about its result;
- where the real state can be verified;
- which events have already been processed;
- one next step;
- the decision without which you can't continue.

This isn't bureaucracy for the joy of bureaucracy. It's a replacement for the sentence, "surely it remembers." It doesn't? You have a record. It does? You still have a record, so you can check its claim.

For Z-0002, the handoff could look like this:

> Continue on host Home-PC in folder [path] with job Z-0002. It is confirmed that quote-v1.md for CZK 2,000 was sent to the fictional customer. quote-v2.md is supposed to be a draft for CZK 3,600 based on clarification.md; after the failure, we don't know whether the file was created. Don't automatically repeat its creation. First verify the existence and contents of v1, v2, and the source note. Show the confirmed state, differences, and one next step. Don't send anything, and don't overwrite any existing file without my confirmation.

If v2 exists and is correct, you continue with the review. If it exists and is wrong, you fix the specific error. If it doesn't exist, only then do you have it created. Three different states, three different responses. One blind repeat would be convenient for about the first two seconds.

There's also a fourth state: v2 exists, but you don't know what source material it came from. You don't declare that version correct just because the numbers add up. Compare it with `clarification.md`, the price profile, and the history of v1. Recovery isn't a hunt for the newest file. It's a return to the last documented state.

Same Work, Different Working Copy

When returning from your phone, return to the same task on the connected computer. If you open a copy of the folder on another computer instead, check what's actually in it. The same project name doesn't guarantee the same files. Quote v2 may have stayed at home while the laptop shows you only v1.

Before a move like that, finish any write in progress and note the last confirmed output. On the second computer, don't just say, "carry on where you left off." Verify the target folder and the actual files. If you can't tell which copy is authoritative, return to the original host. More copying won't settle that conflict by itself.

When to Bring In a Helper

A helper makes sense when it gets a separate piece of work with its own result. One can verify the current documentation. Another can check the quote calculation against the price list. Another can read the finished draft as a customer. Each one needs to know what it may read, what it may change, and how you'll recognize completion.

It makes no sense to unleash three agents to edit the same quote, the same tracker, and the same invoice at the same time. That isn't a team. That's three people with one pen and no agreement.

The number of models isn't a quality metric. One is enough for a simple step. Two may be useful for two independent checks. Three models for every task isn't a law, and it certainly isn't proof that the result will be correct. Divide the work by boundaries and risk, not by how many colorful names fit in a diagram.

> Check only [specific output] against [specific sources]. Don't change anything and don't repeat any external action. Return the differences you find, links to evidence, and uncertainties the sources can't resolve.

The helper returns its review to the main task. The main task decides what gets written. The moment two helpers own the same file or the same external action, you've set the boundaries badly.

Recovery Must End With One Step

After recovery, you don't need a grand new plan. You need one sentence: "V2 exists, the numbers match, v1 remains, and my next step is to review the text." Or: "We don't know whether the invoice was created; the next step is to search for it in the application by customer and amount."

Until you can write that sentence, you haven't recovered from the failure. You're merely running again.

9. Another Customer Will Turn Up Tomorrow.

One good quote isn't a system. It's one good quote.

A system starts when the next customer turns up and you don't have to work out all over again where your price list is, what may be sent, and how to identify the latest version. It doesn't mean forcing every job into the same template. It means collecting facts, preparing a draft, checking risk, and saving the result in the same way each time.

Take the Process, Not Someone Else's Business

Turn the whole book into your own short process:

- I save the new source material for the job and give it one ID.
- I check the business profile, prices, and rules that apply to the work.
- I have a new output prepared instead of rewriting history.
- I open the actual file and verify the amounts, scope, date, and approval status.
- I approve a binding action separately.
- I record the actual result in a single tracker and set the next step.

A graphic designer will check different facts from a carpenter. A service technician will have a different worksheet from a consultant. That's fine. The structure stays; you adapt the content. If you also copy details that make no sense for your work, you aren't building a system. You're just carefully reproducing someone else's mistake.

Measure Time Until You Have a Result You Used

"AI saved me two hours" sounds great. Where did that number come from?

If the answer is "it just sort of felt that way," you don't have a measurement. You have a feeling with a decimal point.

For several similar jobs, record the time from receiving the source material to reaching a result that you actually checked and could use. Include preparing the sources, writing the task, waiting, corrections, review, and any manual finishing work. Alongside that, note:

- how many drafts passed without a major rewrite;
- how many errors you had to correct;
- whether a duplicate version or action was created;
- whether the output was actually used;
- how long comparable work took with your previous process.

Compare similar cases. The first run may be slower because you're building the profile and templates. The tenth may be faster. Don't choose one easy quote and compare it with one old disaster just to make the chart look good. That's how you manufacture marketing, not knowledge.

The record doesn't need to be a scientific laboratory. For every completed job, the date, type of work, start time, time the usable version was finished, number of corrections, and a short note about what caused delays are enough. Separately mark time that won't recur next time, such as creating the profile for the first time. Don't quietly subtract it. Show it alongside the normal time. After three to five similar cases, you'll see more than you did after one exciting evening.

The benefit doesn't have to be time alone. You might finish the quote the same day, catch a missing detail, preserve the change history, or stop forgetting unfinished orders. Those are useful results. Just don't invent a monetary value for them until you can support it with evidence.

Let Astra Work on More Things Only After That

Once one job is finished, AI will happily suggest another ten projects. A new CRM. Marketing automation. Inventory. A dashboard. And ideally an agent to manage the other agents, because apparently you can't even sell a bread roll these days without a diagram.

Stop it.

> Don't start a new project. From the job just completed, extract only the repeatable process, the sources used, the errors found, and one template change for next time. Don't implement anything else. Show me what to preserve and what to change before the next customer.

First, finish one thing for the second time. Then for the third. Watch whether the number of corrections falls and whether the result makes it into your work faster. If it does, you have a benefit. If it doesn't, adjust the process, narrow the task, or don't use AI for that part.

Every so often, read your own process without Astra. Are the prices current? Do the approval boundaries still make sense? Has one customer's exception lingered there while pretending to be a general rule? AI can help you find the changes, but the decision is yours. Otherwise, you'll be very efficiently repeating a rule that's been out of date for six months.

That last option isn't defeat. Astra doesn't have to do everything to be valuable. It should do specific work that you can take over, use, and measure honestly.

Appendix: what to open and what to check

The book is free. The model, its usage limits and any invoicing service aren't made free by that. You don't need to buy another book from us to use these exercises. This is educational material, not accounting or legal advice for your business. For actual rates, settings and permissions, go back to your real records and the appropriate professional, not our fictional furniture assembly job.

This English edition keeps the original Czech example and its amounts in CZK. We haven't converted prices into dollars or turned a Czech invoicing example into a promise of compliance in your country. Fakturoid is the example provider, not a requirement to use that service. The English prompts and localized exercise files have not completed their own live model trial.

Twelve templates alongside the book

In `exercise-pack/templates`, you'll find:

- `01-business-profile.md`: what you do, what you charge and your sources.
- `02-workspace-and-sources.md`: where the current records live.
- `03-inquiry.md`: a record of the customer's request.
- `04-quote.md`: a draft quote, line items and approval.
- `05-customer-voice.md`: two good examples and limits on tone.
- `06-job-sheet.md`: what to prepare before doing the job.
- `07-job-tracker.csv`: an empty job tracker for your own work.
- `08-change-request.md`: a change request, its impact and confirmation.
- `09-invoice-preparation.md`: what actually belongs on the bill.
- `10-invoices-and-payments.md`: issuing, sending and payment kept separate.
- `11-mobile-task.md`: the host computer, job and exact output.

- `12-recovery-and-handoff.md`: where work stopped and what's still unknown.

These aren't a fully populated business system. Fill in the template you need right now. You can rename things in your own business, but your prompts and other references must then use those same names. Keep the supplied paths for the first exercise. Import CSV as UTF-8 with a semicolon delimiter. If everything lands in one column, that's not a new business structure. It's a bad import.

What we mean by tested

The original two runs, each covering the business overview and the quote, used the same inputs in separate copies. Two further runs tried job preparation in three trades and the subsequent tracking of three jobs. The graphic-design copy needed a less generic rewrite; the correction and first drafts were retained. Reprocessing an event in the tested tracker didn't create another payment. Those are limited practical results, not a measurement of universal reliability. They refer to the recorded Czech trials, not a new test of this translation.

Checking file structure means the paths, identifiers and prepared amounts agree. It doesn't mean the model will understand every customer correctly. A revised prompt hasn't automatically been retested because a similar one once passed. The exercise README records the current scope of verification.

This free public working edition still has open tests: a complete beginner walkthrough, installation on a clean computer, practical invoicing and the Remote workflow. It's a complete working text to try, not a fully tested final manual. A finished PDF doesn't close that gap.

Your short acceptance check

Open the blank protocol in `exercise-pack/checks/test-protocol.md`. Record the actual date, application, selected model and inputs it received. For every attempt, keep the output, one concrete check and any correction. Don't feed it the answer key before testing it.

For quote Z-0001, expect CZK 4,000 without an invented appointment. During job preparation, the CZK 600 shelf stays separate. The morning tracker has three unique jobs and one documented fictional payment of CZK

6,000. In the separate offline exercise, a CZK 2,000 payment against a CZK 4,000 test document leaves CZK 2,000 outstanding. The new Z-0002 quote for CZK 3,600 must not overwrite the sent v1 for CZK 2,000. You don't need admiration here. You need a match with the source.

If something fails, write down what failed. If it wasn't tested, write not tested. "Probably fine" doesn't fit in the verification box, even if you make the font smaller.

Me and Stupid Questions: Astra's on Shift

Will Astra run my business for me?

And what will the customers be paying you for? Attendance?

Do I have to know what I want from it?

No. Then you can both be surprised.

Why is it asking what I charge?

Its fortune-telling licence hasn't come through yet.

It said done. So it's done, right?

I'll call you a millionaire. Did that help?

Can I give it all my files?

Sure. Give the new temp your bedroom keys while you're at it.

Why didn't it send the quote?

Because you wrote "don't send it". Sometimes that happens when someone listens.

If I switch on six agents, do I get six times the finished work?

You might get six unfinished drafts. Still a number.

What if I don't like the result?

Tell it. You're not married yet.

The computer's off. Why isn't it working?

Have you tried sending it a motivational quote?

Can it just agree with everything I say?

Yes. Great question. Excellent direction. Do you approve the next agreement?

Sources and edition

Instructions checked against documentation reflect the Czech source edition's review on 12 September 2026. These links lead to official sources; feature availability and interfaces can change. The fictional jobs use a separate exercise calendar. Translation does not constitute a new live test.

- [S1] OpenAI: desktop app quickstart.
- [S2] OpenAI: models and reasoning settings.
- [S3] OpenAI: Windows app and permissions.
- [S4] OpenAI: GPT-6 Astra model.
- [S-R1] OpenAI: Remote and pairing.
- [S-R2] OpenAI: remote connection requirements.
- [S-F1] Fakturoid: creating an invoice, Czech documentation.
- [S-F2] Fakturoid: sending an invoice by email, Czech documentation.
- [S-F3] Fakturoid: recording a payment, Czech documentation.
- [S-F4] Fakturoid: partial payments, Czech documentation.
- [S-F5] Fakturoid: document drafts, Czech documentation.

The author is Kristian. This text was developed with AI assistance from his experiences, feedback and practical trials, and this English edition was translated with AI assistance. The fictional assembly, design and manufacturing jobs are not customer testimonials for KK make IT. Neither OpenAI nor Fakturoid publishes this guide. Mentioning them does not imply their endorsement or guarantee.

Free public working edition 1.0, English, for readers to try; not a fully tested final manual. The PDF, readable source and editable exercises form one package. A different model or a newer file doesn't prove that the same test has passed. This translation retains the Czech setting and CZK amounts, not country-specific invoicing advice for an international audience.

Report an error to order@kkmakeit.com. Include the book version, chapter or filename and an anonymized description of the discrepancy. Don't attach passwords, access keys or entire customer documents.