A GUIDE BY ORCHA · MAY 2026

AI in the Finance Function.

How finance teams choose, deploy, and use AI day to day.

FOR LEADERSHIP & FINANCE TEAMS IN THE MID-MARKET · EDITION 1 · EN

CONTENTS

What's in this guide.

·	Foreword	3
01	AI creates value on two levels	4
02	The right process makes the difference	6
03	Your data doesn't need to be clean first	8
04	Build or buy	9
05	AI-native or add-on	12
06	AI costs behave like energy costs	14
07	Data protection and governance	15
08	From reactive to proactive AI	16
09	The first 90 days	17
A1	About Orcha	18
A2	Should you build it yourself?	19
A3	Mini-glossary & sources	20
A4	Prompt templates	21

Distilled from 35 posts by the Orcha Community. You can read this guide in any order — the chapters build on one another but also work on their own.

FOREWORD

A short guide to getting started.

The AI market moves fast — new models, tools, and promises almost daily. In all that noise, it's hard for a finance team to tell what actually matters.

The opportunities are real and large. Yet so far only a few teams get past first experiments to something that genuinely works day to day. What usually trips them up are the decisions beforehand: which process first, build or buy, which vendor, what the long-term costs will be.

This guide pulls together our 35 posts from the Orcha Community on these questions — compact, and meant as a starting point. It's no substitute for a consulting project; it organises the questions you have to answer anyway — whether you ultimately choose Orcha, a competitor, or building it yourself.

Nine chapters. From concept to rollout. At the end of each chapter, a question to test your own situation.

— The Orcha Team

AI creates value on two levels.

On one, every individual becomes faster. On the other, agents take over entire processes. Both are worthwhile, but the leverage differs.

The first level is individual productivity: Excel formulas, presentations, contract analysis, email drafts. These are tools like ChatGPT, Copilot or Claude that help individuals with their work. This level is available immediately and is the natural starting point — the whole team can begin within a few days — see the quickstart below.

The second level is process automation: agents that handle incoming invoices, bank reconciliation or the monthly close end to end. This is where the greater potential lies, because these processes largely run on their own and take work off an entire team, rather than only making a single person faster. It is also more demanding: it needs more preparation and domain knowledge before it runs reliably. This guide focuses mainly on this second level.

QUICKSTART FOR PERSONAL PRODUCTIVITY

Four things a finance team can start with this week

1. **Excel:** Claude for Excel or the Excel agent in Microsoft 365 Copilot work directly in the spreadsheet — explaining formulas, building models, commenting on variances.
2. **Presentations:** For slide decks, reach for the agentic mode (Claude Cowork, the PowerPoint agent in Copilot) rather than a simple chat — it builds entire presentations from your data.
3. **Across apps:** Desktop agents such as Claude Cowork access local files and complete multi-step tasks across Excel, Word and Outlook; Microsoft offers similar capabilities in Copilot.
4. **Data protection first:** Only Business and Enterprise accounts are GDPR-compliant. With those, your data isn't used for training, and Zero Data Retention can be switched on depending on the contract. Personal accounts are not suitable for business data (more in Chapter 07).

From the field: People who build AI firmly into their daily work become far more productive than those who reach for it only now and then for one-off questions. A proven approach is to name two or three internal champions who use AI intensively and pass on what they learn to the team — for instance in a short all-hands or a hackathon. With a little guidance, adoption usually comes quickly, and productivity rises across the whole team.

AI is a teammate.

Use AI like a search engine and you usually get superficial results. Treat it like a new teammate on their first day — capable, but without context — and you generally get markedly better ones.

Modern AI (an LLM) is a very capable teammate on its first day. It has read a great deal, thinks quickly, and works around the clock. But it doesn't know your chart of accounts, your clients or your internal terms. Without that context, what you get back is usually unusable.

A bare request produces generic material. Add a paragraph of context — who you are, what your company does, what data you have — and it becomes a draft the team can build on. The result depends less on the model than on how well you describe the task.

The habit that matters

The greatest leverage often lies less in the "right tool" than in how you work: *before* every task, ask once whether AI can help — not only once you're stuck. Make this a habit and you capture a large share of the productivity gains on offer. As a rule of thumb: AI first, manual work only where it can't get any further.

Over time, this also shifts the team's role — away from processing, towards judgement. The work then consists less of capturing documents than of deciding which cases need attention and where the AI must be corrected.

EXAMPLE · TWO APPROACHES, SAME QUESTION

In controlling, someone asks: "Write me a variance analysis for Q2." Output: generic, full of Excel tips, with no connection to the business.

Second attempt with context: "We are a machinery manufacturer, €80M in revenue, SKR 03. Attached: our Q2 plan-actual and the previous quarter's variance analysis. Compare the two quarters, explain the three largest variances with their probable drivers, and phrase them so the supervisory board understands them without follow-up questions." Result: three clear drivers, with follow-up questions for the cost-centre owners.

SELF-CHECK

Three indicators of "AI as default"

1. You open Claude or ChatGPT *before* you open Excel.
2. You let the AI draft emails rather than typing them yourself.
3. When the AI is occasionally unavailable, you take a break rather than working manually.

The right process makes the difference.

Before you compare tools, settle a different question: which process in your finance team actually deserves to be automated? A 2×2 matrix and a 15-minute test are all you need.

With that, attention shifts from the first level — individuals working faster with AI — to the second: processes that an agent takes over for the whole team. And it starts with picking the right process.

Choosing the process matters more than the technology. Complex tasks with many edge cases and a great deal of judgement rarely make good candidates at first; a structured process that recurs monthly usually works far better.

Two conditions must be met *at the same time* for a process to be suitable for AI:

- **Time spent:** More than roughly 5 hours per week across the team.
- **AI quality:** A 15-minute test with real data and documents shows how well AI can already handle it today — ideally with the provider's strongest model and a detailed, well-written prompt. If you have to correct the AI repeatedly, the task is usually not suitable.

The selection matrix

The quick exercise: first note down every process you can think of, and place each one along these two dimensions. The processes in the "Start now" field are picked up again on the next page.

	AI QUALITY LOW	AI QUALITY HIGH
TIME HIGH	**Watch.** Lots of pain, but AI not yet ready. Re-test every six months.	**Start now.** High pain, AI ready. This is where the greatest leverage lies.
TIME LOW	**Do not prioritise.** No leverage, no maturity. Ignore.	**Quick win.** Small but reliable. Helps the team build confidence.

Highlighted in colour is the field with the greatest leverage: high pain, AI ready — start here first.

Six questions that show whether a process is AI-ready.

Ask these six questions for every process you placed in the top-right field of the matrix. Five or more "yes" answers: a clear candidate. Three to four: worth a closer look. Fewer than three: leave it manual.

1. **Time spent:** Does the process take the team more than 5 hours per week?
2. **Repeatability:** Does it run regularly in a similar form (daily, weekly, monthly)?
3. **Passes the practical test:** Does the AI produce a usable result from real documents within 15 minutes?
4. **Error tolerance:** AI will not get every case right. Good candidates are processes where the occasional mistake is bearable — or where a human signs off on the result at the end anyway.
5. **Data availability:** Can you get hold of the data you need — via an integration, an API or a simple upload? Whether the data is structured does not matter here; the AI takes care of that (more in Chapter 03).
6. **Data protection settled:** Is it clear which data the AI may see and which it may not?

EXAMPLE · AP AUTOMATION

Incoming invoices at a mid-market company: 500 invoices a month, 12 to 15 minutes each by hand — over 100 hours across the team. All six questions answered "yes". This belongs in the top-right corner.

Counter-example: a four-way match with many edge cases, or a cash flow forecast with too many variables. On paper they look like candidates — in practice the long tail of exceptions eats up every efficiency gain.

A QUESTION FOR YOUR TEAM

Which three processes would you place in the top-right corner today?

If your team cannot answer this question in 15 minutes, run a one-week stocktake: 3–4 team members log their recurring tasks along with the time they take.

Your data doesn't need to be clean first.

This requirement comes from the world before LLMs — today it holds back projects that should have been running long ago.

The old logic goes: data quality first, then automation. With traditional systems built on rigid schemas, that held true. With AI-based systems it usually no longer does — and the assumption needlessly stalls projects that could already be under way.

Modern LLMs read "Travel exp.", "Travel expenses" and "Reisekosten" as the same category. They handle mixed languages, varying formats and master data that has been kept inconsistently. Data quality still matters. But it comes from *the AI itself*, not from cleaning everything up in the ERP first.

AI as a structuring layer over existing infrastructure

1. **Connect data sources** — ERP, Excel, email, bank portals. No migration needed.
2. **AI reads, interprets, structures** — independent of format and language, with a confidence score per extraction.
3. **Clean data lands in a database** — queryable via SQL, BI tools or further AI.
4. **Existing systems stay** — DATEV, SAP, your chart of accounts. Nothing is replaced.

> EXAMPLE · A $780M CAUTIONARY TALE
>
> A US Fortune 500 retailer halted a planned $780M ERP migration in 2024. Instead, the company layered AI on top of its existing systems (source: McKinsey 2025). This is not just a Fortune 500 trick. A mid-market company running DATEV and years of Excel files can layer AI on top of its existing systems too — without changing a single system or process.

A QUESTION ABOUT THE OLD SEQUENCE

Which AI pilot is waiting on a data migration at your company?

If one comes to mind, look at it as your next pilot candidate.

Build or buy.

Around 31% of internal software projects stay on schedule, on budget and within the agreed scope. For every euro of development cost, expect four to five euros of maintenance over five years (Gartner).

Building standard processes in-house rarely delivers a competitive advantage — it is *routine work that everyone in the market has to do.* Werner Vogels (Amazon) calls this "undifferentiated heavy lifting": effort that does not set you apart from the competition. For a finance team, that usually means buying a ready-made tool and building only what genuinely differentiates you.

This does not mean building in-house never makes sense. With your own code, standard processes can often be improved by 20–40%, in some cases more. But the edge cases you cannot see today soon cost more than an external vendor's software licence. You can digitise invoice intake yourself and reach a reasonable automation rate — but a 90%-plus touchless rate on international documents, plus four-way matching, fraud detection, automatic supplier onboarding and automatic supplier communication, is a software project in its own right.

	BUILDING IN-HOUSE	BUYING
Time-to-value	9–18 months	Weeks
Success rate	31%	Proven product on the market
Maintenance per year	15–20% of initial cost	included in the licence
Further development	Build every new feature yourself	Vendor ships new features continuously
Regulation (GoBD, GDPR)	Your own responsibility	Vendor
5-year TCO	4–5× initial cost, unpredictable	Linear, predictable

Four questions that decide whether to build.

The first two are the Gartner conditions — building in-house is only worthwhile if both apply; the other two come from experience:

1. **Is it core to your competitive advantage?** (Gartner)
   If three competitors have to solve the same problem, it is rarely an advantage — more likely operating overhead that everyone bears.
2. **Is there no suitable solution on the market?** (Gartner)
   Only when the market offers nothing suitable is it worth looking at building in-house.
3. **Do you have the capacity to maintain the system over years — and to implement regulatory changes yourself?**
   One GoBD adjustment a year, one GDPR audit, an API migration every 18 months — each is a permanent draw on the team.
4. **What could your development team build instead — something that generates revenue?**
   Opportunity costs are invisible but real: every hour of development work on an accounting system is an hour taken from a product that customers pay for.

A further risk lies in knowledge held by single individuals: when the key person leaves, you often have to rebuild that knowledge from scratch — unlike with a vendor that keeps the system maintained for the long term.

> WHEN IT GOES WRONG
>
> **Lidl** stopped an in-house SAP replacement project in 2018 after around **€500M** — with no working system. **Nike** lost **$100M** in revenue in 2001 after an in-house forecast system miscalculated inventory levels.

WRITTEN-EVIDENCE TEST

Can you provide written evidence for both Gartner conditions for every planned in-house build?

Five questions every buying plan must answer.

"Buying" calls for its own due diligence too. The licence fee says little about the follow-on costs, and a demo rarely shows what actually works.

1. **Does the vendor adapt to your world — or the other way round?**
   DATEV document transfer and accounting-data service as standard (DATEVconnect for deeper integration), an open API to your ERP (SAP, proALPHA, abas, Sage, Microsoft Dynamics), connection to your chart of accounts, collaboration with your tax advisor. If the answer is "you have to adapt your processes to our tool", the change-management costs are usually a multiple of the software licence costs.

2. **Does the data belong to you — even after you switch?**
   Where is the data held (EU vs. US), who are the sub-processors, and is there a documented export path in a usable format? You need an exit process assured in writing.

3. **Is the system API-first?**
   Precisely because AI tools are evolving so fast right now, you have to avoid lock-in — the priority is open interfaces. More and more AI tools need direct access to your data; an open, documented API makes later extensions easier and reduces your dependence on a single vendor.

4. **Is the system auditable — and is the vendor dependable?**
   Traceable decisions with an audit log and confidence value, procedural documentation for the tax audit, a GDPR data processing agreement, an EU AI Act-compliant risk classification.

5. **What does running it cost beyond the licence?**
   ERP connection, training the team, reworking internal processes, vendor risk (acquisition, insolvency, product discontinuation) — and the most expensive item: lock-in. What does switching cost in five years?

EXAMPLE · CLOSED VS. OPEN SYSTEM

Vendor A is a closed system: data only enters and leaves through the user interface, and a full export is possible "on request" at best. Every extension depends on what the vendor unlocks.

Vendor B is API-first: all data can be read and written through a documented API, plus an MCP (Model Context Protocol) server for direct access by AI tools. Both cost the same — but B fits into your existing systems and keeps the door open for a later switch.

QUESTION FOR THE VENDORS ON YOUR SHORTLIST

"What happens to our data and our integration if we switch in five years?"

At the same time, ask for detailed API documentation — and for a clear list of what can only be done through the user interface but not through the API. A concrete, written answer is what helps most here.

AI-native or add-on.

AI is advancing fast. Anyone buying today needs software built for an AI future. From the outside, it is often hard to tell how strong a vendor really is on AI. At heart there are two patterns: individual AI features bolted on as an add-on — or an AI-native application.

Established vendors have refined their platforms over years and added AI in recent months. AI-native vendors start from the AI models as their foundation. At first glance the two often look the same. The difference is fundamental — and it shapes how well a vendor keeps pace as AI moves on.

A simple test reveals what you are dealing with: picture the system with all its AI features stripped out. If it carries on running on old template logic, AI is only sitting on top; if nothing is left, AI is what carries the system. Day to day, the difference shows up like this:

	TEMPLATE-BASED (CLASSIC)	AI-NATIVE (LLM-CENTRED)
Onboarding	Several months of learning	Works immediately
New supplier	Create a new template	Works immediately
Layout change	Adjust the template	The model recognises it
Foreign language	A separate template per language	The model understands it
Free-text fields	Tedious, error-prone	A strength of the model
Account coding	Maintain rule sets	The model suggests it
Maintenance over time	Rises with suppliers	Falls with volume

A second criterion is model lock-in. The model itself is fast becoming a commodity — the edge lies in the harness around it: workflows, integrations, interfaces, controls. So the useful question to put to a vendor is this: can you swap out the underlying model without rebuilding the whole workflow?

40% recognised doesn't mean 40% less work.

OCR is an older generation of AI: it reads characters, but leans on rules and templates that someone has to maintain. It tells you what it has read — but not how confident it is. So you have to check every document. At 40% correctly recognised, the time you actually save might be 20%.

Modern LLMs are a big step beyond that. They handle cases with no fixed rule set, and they attach a confidence score to every extraction, so you only review what the system is unsure about. The time you actually save climbs to as much as 90% — not 20%.

More than just reading

Modern AI does more than read — it vets suppliers, reconciles invoice, purchase order and goods receipt (four-way match) and suggests the account coding. A good sign: if the AI takes on work that experienced specialists would otherwise do, there is more to it than plain text recognition.

EXAMPLE · AN AI-NATIVE AP SYSTEM

Orcha passes the test with a "No": without an LLM the product would not exist. Not the only AI-native vendor on the market, but a clear illustration of the principle. 80–90% of invoices go through fully coded and posted — with no human rework. The industry average is around 33% (Ardent Partners 2025).

A QUESTION FOR THE VENDORS ON YOUR SHORTLIST

What happens when a new supplier sends an unknown invoice format?

Does someone have to create a template — or does the system recognise it on its own and set up the new supplier automatically in the process? The answer tells you whether AI is the foundation or just helps out in places.

AI costs behave like energy costs.

Twenty years of SaaS have got us used to fixed licence fees. AI changes that: costs rise with usage.

A power user can easily cost several times what a light user does, so a single flat rate often no longer pays off for vendors. Three shifts in the market follow from this:

1. **Flat rates are disappearing** or being capped; in 2026 there is little genuine "unlimited" left among AI-native tools.
2. **Hybrid models are becoming the norm** — a base fee plus a usage-based portion.
3. **Usage transparency becomes a negotiating point** — insist on contractual visibility into actual usage.

The new metric: cost per transaction

Monthly totals say little once the cost base becomes variable. What matters is the cost of a single AI action — one document, one analysis, one file. Wherever you can, base billing on something you can control and forecast yourself, such as per document rather than per token. You know how many documents you process in a month; how many tokens those will run up is almost impossible to predict.

For budgeting, this means that running AI is an operating expense (opex), not a one-off investment. Finance teams already know the move from one-off purchases to recurring fees from the SaaS world; what is new is that the cost also swings with usage.

EXAMPLE · POWER USER VS. LIGHT USER

Power user: Someone in controlling who spends a week building a variance analysis in Excel with the help of AI burns through a lot — intensive, context-heavy sessions add up.

Light user: Someone who turns to AI only for the odd question or draft stays well below that — the gap between the two soon runs to three or four times. The same tool, the same licence, very different usage.

FOUR POINTS FOR EVERY AI CONTRACT

What to lock in at the next negotiation

1. **A clear pricing metric:** per document or per outcome — something you understand and can control yourself, not per token.
2. **Cost caps and alert thresholds:** at what point are you notified? At what point does an automatic block kick in?
3. **Real-time usage visibility:** daily or hourly resolution — not a monthly bill.
4. **A projection on real data:** run the numbers against your actual volumes before signing, and put the expected cost range in writing — it heads off surprises later.

Data protection and governance.

With the right business account, a signed DPA, EU data residency and an internal policy, GDPR-compliant AI use can usually be set up in days.

> **Free and personal AI accounts are usually the wrong choice for a finance team.**

Free and personal accounts (OpenAI Free, ChatGPT Plus, Claude Free) train on your inputs by default. Business accounts (OpenAI Enterprise, Claude Team/Enterprise, Microsoft 365 Copilot) are contractually exempt from this. Switching to a business account removes this risk; the additional cost is negligible.

Four data classes, four approvals

Assign your data to four levels — each level determines which account may see it:

- **Public** — any account, including personal ones.
- **Internal** — only business accounts with a signed DPA.
- **Confidential** — business accounts with EU data residency.
- **Strictly confidential** — only in a self-hosted environment. A dedicated API key (Bring-Your-Own-Key) does not help here: the data still passes through the provider's API.

Then there is the access path — a single upload, a standing project or a fixed connector. The more sensitive the data, the tighter the access should be.

10-POINT CHECK FOR GO-LIVE

Before the first document runs through the AI

- ☐ Signed data processing agreement (DPA)
- ☐ No training on your data — confirmed in writing, or toggled off in the admin console with the major cloud providers
- ☐ EU data residency enabled (where available)
- ☐ Data protection impact assessment documented (DPIA)
- ☐ Internal AI policy adopted
- ☐ Sub-processor list reviewed regularly
- ☐ AI-literacy training carried out (EU AI Act, mandatory since Feb. 2025)
- ☐ Works council informed, if any (co-determination under §87 BetrVG / ArbVG for performance evaluation)
- ☐ GoBD process documentation updated (for tax audit)
- ☐ AT: data protection officer notification reviewed, BAO §§131/132 covered

SHADOW AI · UNOFFICIAL AI USE WITHIN THE TEAM

If the official route is too restrictive or too cumbersome, employees reach for a personal account. A ban rarely solves this — what helps is an official route that works at least as well as the unofficial one.

From reactive to proactive AI.

Reactive AI waits for prompts. Proactive AI watches the data flows continuously and speaks up before anyone asks.

Three typical processes show what this means in practice:

PROCESS	REACTIVE	PROACTIVE
Accounts payable (AP)	The invoice is captured and posted.	AI flags duplicates and deviations from the purchase order before posting.
Purchase order & budget	The overrun shows up in the month-end close.	AI raises a warning at the requisition stage, as soon as it would exceed the budget.
Suppliers & procurement	Terms are reviewed at contract renewal.	AI continuously spots savings opportunities and concentration risks.

THE 5 MATURITY LEVELS · FROM CHAT TO AUTOPILOT

STAGE 01
Session
Individual chats. Output: ~2-3× faster.

STAGE 02
Skill
Routines made repeatable. ~3-5×.

STAGE 03
Workflow
Steps chained together. ~5-10×.

STAGE 04
Agent
Runs autonomously. ~10-25×.

STAGE 05
Agent swarm
Several agents in parallel. 25× and more over the long term.

EXAMPLE · WHAT WE ARE BUILDING AT ORCHA

At Orcha, agents run around the clock: capturing documents, flagging deviations, surfacing cashflow signals early. Stage 4, in places Stage 5 — one of several systems on the market actually running in production.

FINAL QUESTION

Which stage is your team at today — and what is the next concrete step?

We often see two patterns: those who force Stage 5 straight away drown in complexity; those who stay at Stage 1 miss that Stage 2 is only one skill away.

The first 90 days.

The previous chapters lay the groundwork — this roadmap turns it into a concrete approach that takes you from the mandate to a tangible project plan.

Week 1 — Define the mandate

A one-page mandate document: which process (exactly one) is the candidate? What will be measured over 90 days? Who is the driving force from your team, who from IT, who from audit/legal? What is explicitly out of scope? This page later becomes the progress report to senior management. Regardless of the pilot, it is worth securing the biggest quick win straight away: give the whole team access to an AI assistant and train them on it repeatedly.

Week 2 — Set up the pilot

Set up the pilot on one process: 30 sample documents, data-protection check signed before the first record. Define the success metric before the first document: how many documents go through without manual handling, and how long does the rework take? What matters most here are the numbers from your own data, far more than the demo.

Weeks 3–4 — Decide, connect, train

Choose the provider based on the pilot's numbers, not on the presentation. In parallel: sign the DPA, activate EU data residency, train the team, adopt the internal policy, send the process documentation to the tax advisor. The go-live check from Chapter 7 belongs before the first production document, not after it.

Weeks 5–13 — Go live, measure, expand

First process in production, with human-in-the-loop at low confidence. In the weekly review:

- Which exceptions come up?
- Which of these exceptions can be captured as a skill?

After 90 days comes the progress report — cost per transaction, time gained for the team, the next prioritised process.

> EXAMPLE · 90 DAYS AT A MID-MARKET COMPANY
>
> A machinery manufacturer with 80 employees started with AP in February. Day 14: mandate document signed. Day 30: pilot tested with real documents. Day 60: in production with human-in-the-loop. Day 90: first progress report. 78% of invoices go through without rework. The finance team gains back two days a month.

THREE THINGS AFTER 90 DAYS

What you can show senior management

1. **One process in production**, with a confidence-based workflow.
2. **Three metrics with comparison values** — before / after.
3. **A prioritised roadmap** for the next three processes.

Orcha — AI-native finance software for the mid-market.

Orcha builds finance software that wouldn't have been possible before large language models. Documents, invoices, reconciliations and reports run largely automatically — across existing ERP systems, without migration.

The platform covers the day-to-day work of the finance department: accounts payable and receivable, document management, approval workflows, analysis and reporting. Every document is captured down to the line-item level — with GL account, cost centre, tax rate and a confidence score that shows where someone should take another look. The figures are available in real time, not just at month-end.

Orcha connects to your existing systems: DATEV, ERP and SAP environments, Excel, BI tools, email, Teams and Slack, plus an open API. Data protection and traceability are built in — GDPR-compliant, GoBD-compliant, with an end-to-end audit trail.

This guide exists because many finance teams already know the right tools, yet are often left on their own when it comes to deciding. The tools in this guide work whether or not you ultimately choose Orcha.

Learn more

getorcha.com — product, demo, case studies, further posts from the Orcha community.

Edition 1 · May 2026 · Distilled from 35 Orcha community posts. Free to share. Sources in the appendix.

Three AI assistants, three use cases. Code is often unnecessary.

Management and finance leaders don't need to code. What helps is knowing *when* an AI assistant like Claude can handle a process directly, when a no-code skill is enough, and when a coding assistant such as Claude Code is worth it.

In 2026, "building it yourself" rarely means "building software" anymore. The three realistic tiers are:

- Ask an AI assistant like Claude directly.
- Build a skill that makes the process repeatable.
- Use a coding assistant like Claude Code — for something that runs on its own.

Code is the most complex of the three tiers, and it is often overrated. The reason: a local tool is quick to build, but the effort comes afterwards. As soon as a team starts using it, access rights, an audit trail for changes, version histories and data protection all come into play — and the quick prototype turns into a software and maintenance project. This part is especially easy to underestimate without a technical background, which is why code is rarely the right choice for most everyday cases.

TIER	WHEN IT MAKES SENSE	EXAMPLE
Claude directly Chat or Excel add-in	One-off or infrequent tasks, ad hoc analyses, briefings, document review.	Contract analysis with an IFRS 16 check, variance commentary in the quarterly report.
Skill / project No-code SOP	Recurring tasks that the whole team should handle in the same way.	Monthly reporting commentary to a fixed template, standardised supplier onboarding checklist.
Claude Code "Vibe Coding"	When a standalone artefact is to be created: a dashboard, a script, a small tool.	Budget-vs-actual dashboard with filters, a reconciliation script for 200 entries.

DECISION RULE

Without a technical background, building your own code almost never pays off.

Anyone who handles the same task repeatedly saves time with a skill — and no-code skills today cover a great deal, right up to schedules and connectors in the cloud. Your own code, by contrast, rarely pays off as long as no one on the team can maintain it over the long term.

Mini-glossary & sources.

Eleven terms used in this guide

LLM (Large Language Model)
The model behind Claude, ChatGPT or Gemini. Generates text from statistical patterns — not through database queries.

Token
Unit of text that determines AI costs. 200 pages ≈ 80,000 tokens.

Context window
How much text the AI can read "at once". Today this is enough for entire contracts or quarterly reports.

Prompt
The instruction given to the AI. Better prompts usually produce better results — often more important than the model you choose.

Hallucination
When the AI gives a plausible-sounding but wrong answer. Not a malfunction, but a property of the model. Always verify critical figures.

RAG (Retrieval-Augmented Generation)
A method that lets AI consult your internal documents before answering.

Agent
AI that independently carries out several steps — it does not just answer, it acts.

Confidence score
A measure of how certain the AI is about an extraction. The basis for workflows with a human review stage. Threshold configurable per use case.

MCP (Model Context Protocol)
An open standard that lets AI tools access your systems in a controlled way — reading and writing data.

Zero Data Retention
A contractual assurance that the provider does not store your inputs. Can be enabled on Business and Enterprise accounts depending on the contract.

Vibe Coding
Software is created by describing it in plain language; an AI coding assistant builds it. Good for small tools, not for load-bearing systems.

Sources

Standish Group (success rate of internal IT projects) · Gartner (TCO, build-vs-buy rule) · Ardent Partners 2025 (AP automation rates) · McKinsey 2025 (ERP migration vs AI layer, monthly close) · APQC, Billentis, PYMNTS, IOFM (AP benchmarks) · Bessemer (AI margins) · Anthropic Wall Street Prep Benchmark (Excel modelling) · BaFin / FINRA 2026 (governance) · EU AI Act (in force Feb 2025, high-risk enforcement Aug 2026).

All 35 original Orcha-Community posts: getorcha.com/community

Six templates a finance team can use tomorrow.

You can use these templates straight away: add your own data and context, then replace the [placeholders]. They work in any mainstream AI assistant and in the Excel integration.

01 · PAYMENT REMINDER WITH EARLY-PAYMENT DISCOUNT

"You are our accounts payable (AP) team. Draft a friendly but firm email to [supplier] about invoice [no.] for [amount], now [X] days overdue. Note the payment due date [date] and the early-payment discount available if paid by [date]. Tone: respectful and solution-oriented; keep the business relationship intact. Max. 120 words, with a subject line and a clear call to action."

02 · EARLY-PAYMENT-DISCOUNT OPTIMISATION

"You are our treasury analyst. Attached: a list of open payables with early-payment-discount terms and due dates. Given a bank interest rate of [X]%, work out which invoices are worth paying early to capture the discount. Output a table sorted by annualised return (invoice, discount rate, return, recommendation) and state the total discount gained."

03 · VARIANCE ANALYSIS

"You are our FP&A analyst. Attached: a plan-vs-actual comparison for Q[N] by cost centre. Identify the five largest variances (absolute and in %), explain the likely drivers behind each one, and flag what is one-off versus structural. Close with three pointed follow-up questions for the cost-centre owners."

04 · CONTRACT ANALYSIS · IFRS 16

"You are our technical accountant. Attached: a lease agreement. Assess whether it must be capitalised as a right-of-use asset under IFRS 16. Output: the relevant clauses with references, the term and renewal options, your classification with reasoning, and any open points for clarification. Flag every assumption explicitly."

05 · SUPPLIER RISK SCORE

"You are our procurement analyst. Attached: our top 20 suppliers with revenue shares and categories. For each supplier, rate concentration risk, ease of replacement in the market, and geographic risk on a scale from 1 (low) to 5 (critical), with a brief rationale. Output a sorted table and highlight the three most critical dependencies."

06 · SUPERVISORY-BOARD BRIEFING

"You are our chief of staff. Attached: the monthly close for [month]. Draft a two-page supervisory-board briefing in this order: P&L → balance sheet → cash flow → the three most important topics of the month. Keep the tone factual and concise, with the variance to plan in one sentence per section. Use full sentences rather than strings of bullet points."

A good prompt often achieves more than a stronger model. The templates work with Claude, ChatGPT, Gemini or Copilot — but only with real context and on your own data. More templates in the post "AI shortcuts & tips" on getorcha.com/community.