



A NEBIS FIELD GUIDE

The AI Operating Playbook

For Singapore businesses: how to put AI to work without putting your compliance at risk.

FOR OWNERS, FOUNDERS & FINANCE LEADS

Choose the right work, design workflows people can trust, stay inside Singapore's data and record-keeping rules, and know within thirty days whether it was worth doing.

THE OPERATING IDEA

AI creates durable value when it is attached to a real business outcome, bounded by clear rules, and improved through feedback. Everything else is a demonstration.

Why this exists

Most guidance about AI for small businesses is written by people who advise on AI. This one is written from the other side of the problem.

We rebuilt a regulated professional practice around AI systems — under a statutory framework that punishes carelessness, in a jurisdiction that expects records to be produced on demand years after the fact. That forced a set of questions that a pilot project never has to answer. What happens when the output is wrong and someone has already relied on it? Who is accountable when the machine did the work? What do you show an inspector? Which decisions can never be delegated, no matter how good the technology gets?

This playbook is what we would have wanted at the start. It is deliberately platform-agnostic: it names capabilities, not products, because the tool you choose this year will not be the tool you use in three years, and the operating decisions are what survive.

It is not a technology catalogue and it will not tell you which subscription to buy. It will tell you how to choose the right work, design it so it can be trusted, keep it inside Singapore's data and record-keeping rules, and know within thirty days whether it was worth doing.

How to use this playbook

Start with the business problem, select one workflow, and make the smallest safe change that can produce evidence. Treat every experiment as a learning loop: define the outcome, test the workflow, review the result, and decide whether to adopt, revise, or stop.

Read it once end to end. Then use Chapters 2, 3 and 10 as working tools, and return to Chapter 7 whenever you are about to widen the scope of something.

START HERE

If you only have one hour: complete the Opportunity Scan in Chapter 2, choose one low-risk workflow, write the Success Contract in Appendix C, and put a 30-minute review in the calendar with the person who owns that workflow.

Contents

The AI operating model

Find the work worth improving

Prioritise opportunities

Design reliable workflows

Instructions as specifications

Data, tools and the Singapore layer

Governance, risk and the accountability line

Adoption, measurement and improvement

Eight ways this goes wrong

Practical checklists

A 30-day implementation roadmap

Appendix A — The readiness scorecard

Appendix B — Starter task specifications

Appendix C — The Success Contract

1. The AI operating model

AI should be managed as an operating capability: a set of choices about work, information, decisions, controls and learning. Treated as a tool purchase, it produces enthusiasm and then disappointment. Treated as an operating model, it produces compounding advantage — because the second workflow is far cheaper to build than the first, and the tenth is nearly free.

The model below keeps experimentation connected to execution.

LAYER	CORE QUESTION	TYPICAL OUTPUT
Value	What measurable outcome matters?	Outcome statement and baseline
Work	Which recurring task or decision is constrained?	Workflow map and pain points
Intelligence	Where can AI assist, generate, classify, predict or retrieve?	Use-case design
Control	What must remain human-reviewed or restricted?	Risk tier and guardrails
Learning	How will we know it is working?	Metrics, feedback and review cadence

Most failed attempts skip straight to the Intelligence layer. Someone sees a demonstration, buys a licence, and starts looking for a problem to attach it to. The work above and below that layer is where the value actually sits.

The five operating principles

Outcome before novelty. Begin with time, quality, revenue, service or risk — not with a model feature. "We want to use AI" is not an objective; "our invoices go out four days after the job finishes and we want it to be one" is.

Human judgement where consequences are high. Automate preparation before you automate irreversible decisions. Preparing a payment run is a different risk from releasing one. Drafting a reply to a customer complaint is a different risk from sending it.

Small, observable steps. Make work legible so people can inspect inputs, outputs and exceptions. If nobody can see what happened, nobody can fix it, and nobody will trust it.

Data minimisation. Use only the information required for the task, and protect it throughout. Most data incidents in small businesses are not attacks — they are convenience.

Feedback is a product input. Corrections are the most valuable thing your team produces in the first ninety days. Capture them and turn them into better examples, rules or training. A workflow that nobody corrects is either perfect or unexamined, and it is rarely perfect.

THE MANAGER TEST

Can the person who owns the workflow explain, in one minute, what the AI does, what the human does, and what happens when the answer is wrong? If not, the workflow is not ready to scale — regardless of how well the demonstration went.

2. Find the work worth improving

The best first opportunities are **frequent, bounded, measurable and annoying**. Avoid starting with the most strategic or the most sensitive process. You are not trying to prove that AI is powerful; you are trying to earn enough trust to get a second attempt.

Where it usually pays in an owner-managed business

Small businesses tend to look for AI in marketing, because that is where the demonstrations are. The durable wins are usually further back, in the parts of the business nobody photographs.

WHERE THE WORK SITS	TYPICAL CANDIDATES	WHY IT WORKS
Money in	Quotation drafting, invoice preparation, receivables follow-up, credit-control correspondence	High frequency, template-shaped, directly tied to cash
Money out	Supplier invoice extraction, expense coding, purchase-order matching, payment-run preparation	Structured inputs, clear right answer, easy to check
Records	Bank reconciliation queries, month-end schedules, document classification and filing, variance explanations	Repetitive, evidence-based, currently eats an accountant's week
Customers	Enquiry triage and routing, first-draft replies, order and request extraction from email	Volume-driven, tone matters, human keeps the send button
People	Policy questions, onboarding checklists, meeting-to-action summaries, drafting job descriptions	Frequent, low-stakes individually, quietly expensive in aggregate
Compliance	Preparing the document pack your accountant or auditor asks for, keeping the filing calendar, first-pass contract review	Deadline-driven, painful annually, entirely preparation work

Notice the pattern in the last column of every row: **AI earns its place where the work is preparation**. Preparation is high-volume, low-judgement, and expensive precisely because a person has to do it before the valuable thinking can start. Give the preparation away and the judgement gets faster and better, because it happens against complete information.

The Opportunity Scan

For each candidate, capture evidence — not opinion.

SIGNAL	QUESTIONS TO ASK	EVIDENCE TO CAPTURE
Frequency	Does this happen daily or weekly?	Volume per week; cycle time
Friction	Where do people copy, reformat, search or wait?	Steps; handoffs; rework
Value	What improves if this is faster or better?	Cost, cash, conversion, service, quality
Feasibility	Are inputs accessible and reasonably consistent?	Sources; formats; owners
Risk	What could go wrong, and who is affected?	Failure modes; sensitivity
Adoption	Will the owner use it in the flow of work?	Current tool; incentives; habits

Workflow observation worksheet

Sit with the person who does the work and fill this in while watching them do it. Do not fill it in from memory in a meeting room — the memory version is always the tidy version.

- Workflow name: _____
- Owner and users: _____
- Trigger and desired outcome: _____
- Inputs and systems touched: _____
- Current steps (aim for 5–9): _____
- Most expensive delay or error: _____
- What a good result looks like: _____
- What must never happen: _____

That last line is the most important one on the page and the one most often left blank. Write it before you write anything else about the solution.

GOOD FIRST USE CASES

Drafting from approved source material · summarising internal meetings · classifying inbound requests · preparing first-pass analysis · extracting fields from documents · generating checklists · routing work to the right person.

BAD FIRST USE CASES Anything that touches a bank payment, a regulatory submission, a person's employment, a price a customer will hold you to, or a legal position. Not because AI cannot help — but because these are not where you want to be learning.

3. Prioritise opportunities

Prioritisation is what stops a long list of interesting ideas becoming a scattered programme that dies quietly in the third month. Score each opportunity on value, readiness and risk. Use relative scores; precision matters less than a shared view.

The V-R-R score

Score each dimension from 1 (low) to 5 (high).

Priority = Value × Readiness × (6 – Risk)

DIMENSION	1	3	5
Value	Nice to have; no owner	Useful improvement	Directly tied to a metric you already watch
Readiness	Messy inputs; unclear process	Known owner; some structure	Stable inputs; repeatable process
Risk	High consequence or sensitive	Moderate impact; review possible	Low consequence; easy to reverse

The multiplication is deliberate. A near-zero score anywhere kills the idea, which is correct: a high-value use case with unusable inputs is not a high-value use case, it is a data project wearing a disguise.

Decision rules

- **Start now** — high value, high readiness, low-to-moderate risk.
- **Prepare first** — high value but low readiness. Fix the process or the data before adding AI. This is the honest answer far more often than people want it to be.
- **Pilot with controls** — moderate value or moderate risk. Narrow scope, mandatory review.
- **Do not automate** — high consequence, unclear accountability, or no practical way to detect failure. The third of those is the one people miss: if a wrong answer looks exactly like a right answer, you do not have a workflow, you have a hazard.

SAMPLE SUCCESS CONTRACT (ILLUSTRATIVE)

For the next 30 days, the customer-service lead will use an AI-assisted draft for the ten most common request types. Success means a materially faster first response with no increase in reopened cases. Every draft is reviewed by a person before it is sent. If reopened cases rise at all, we stop and investigate before continuing.

4. Design reliable workflows

A reliable AI workflow is a controlled loop, not a clever prompt. Design the handoffs and the exception paths first; then decide where AI fits. Teams that do it the other way round end up with something impressive that only its creator can operate.

The workflow loop

1. **Trigger** — what starts the work?
2. **Prepare** — what inputs are gathered, cleaned and authorised?
3. **Assist** — what should AI produce or recommend?
4. **Review** — who checks accuracy, completeness, tone and policy?
5. **Act** — what is sent, recorded or changed?
6. **Learn** — what feedback is captured, and when is the workflow reviewed?

Control points to define

CONTROL POINT	MINIMUM DESIGN DECISION
Input boundary	Which sources are allowed? What information must be excluded or masked?
Output standard	What fields, format, tone and evidence are required?
Human review	Who reviews, at what threshold, and against what checklist?
Exception path	What happens when confidence is low, facts conflict, or the request is novel?
Audit trail	What is logged: input reference, output, reviewer, decision, date?
Stop condition	What signal pauses the workflow for investigation?

Worked example: receivables follow-up

A generic table is easy to agree with and hard to use. Here is the same structure applied to a workflow almost every small business has, and almost none has designed.

The situation. Overdue invoices are chased when someone remembers, usually the owner, usually late, usually in a tone that varies with their mood. Some customers are chased twice; some strategic accounts are never chased at all.

Trigger — an invoice passes its due date, checked daily.

Prepare — pull the invoice, the customer's payment history, any open dispute or credit note, and the agreed terms. Exclude anything not needed: no bank details, no unrelated correspondence.

Assist — draft a follow-up message matched to the situation: first reminder, second reminder, or escalation. Include the invoice reference, amount, due date and the specific next step. **Flag** — do not chase — any account with an open dispute, an unapplied credit, or a payment arrangement.

Review — the person responsible for credit control reads every draft before it goes. Anything above an agreed value, or to a named strategic account, goes to the owner.

Act — the message is sent from a human mailbox, and the contact is logged against the invoice.

Learn — weekly, review which drafts were edited and why. Edits are the signal. If the same edit appears three times, it belongs in the rules, not in a person's head.

What must never happen — a chaser sent to a customer who has already paid; a chaser sent while a dispute is open; a threat of legal or credit action generated without a human decision; a payment instruction altered.

That last item is worth pausing on. Invoice and payment-redirection fraud is one of the most common ways money leaves a small business. Any workflow that touches payment details must treat bank-detail changes as a human-verified, out-of-band step. Always. There is no efficiency argument that survives this one.

Worked example: the month-end pack

The situation. Management accounts arrive three weeks after month end, by which time they describe a month nobody can influence any more.

Trigger — the ledger closes.

Prepare — bank statements, supplier invoices, expense claims, payroll summary, sales listing, and last month's pack for comparison.

Assist — classify unreconciled items; extract and code supplier invoices; prepare the standard schedules; draft plain-language explanations of every variance above an agreed threshold, citing the underlying transactions.

Review — the accountant checks the classification and the reasoning, not the arithmetic. Anything the system flagged as uncertain goes to the top of the list.

Act — the pack is issued with the review recorded.

Learn — track how many items were reclassified on review, and which categories they came from. That number should fall month on month. If it does not, the rules are wrong.

DESIGN FOR EXCEPTIONS

If a workflow only describes the happy path, it is a demonstration, not a process. Before going live, write down at least three exceptions and how each is handled: a missing input, two sources that disagree, and a request with unusually high impact.

5. Instructions as specifications

A good prompt makes the work explicit. It names the role, the objective, the context, the constraints, the shape of the output and the review standard. The useful test is not "did it produce something good?" but **"could a capable new colleague follow this instruction and produce the same thing?"** If not, you have a lucky result, not a process.

The CLEAR pattern

ELEMENT	PURPOSE	EXAMPLE
Context	Give only the relevant background	"You support a 20-person services business in Singapore."
Limit	State boundaries and exclusions	"Use only the supplied policy extracts. Do not use outside sources."
Expected output	Define the artefact	"Return a triage table with reason codes."
Audience	Match depth and language	"Write for a busy team lead, not a specialist."
Review	Specify checks and uncertainty	"Flag unsupported claims and missing data rather than filling gaps."

The reusable template

PROMPT

You are [role]. Complete [task] for [audience] using only [approved inputs]. The desired outcome is [measurable result]. Follow these constraints: [rules, exclusions, tone, length]. Return the result as [format]. Before finalising, check [quality criteria]. If information is missing or sources conflict, state the gap and ask for the smallest clarification needed — do not assume.

That final sentence does more work than the rest of the prompt combined. A system that guesses confidently is the single most expensive failure mode in this entire field, and it is largely an instruction problem.

Specification quality checklist

- ☐ The task has one primary objective
- ☐ The source material and its authority are clear
- ☐ The output format is easy to inspect at a glance
- ☐ The instruction distinguishes facts, assumptions and recommendations
- ☐ The workflow says when a human must review or approve
- ☐ A test case exists with a known expected result

That last item is what separates a process from a habit. Keep five to ten real examples with known correct answers. When you change the instruction, run them again. This takes twenty minutes and is the only honest way to know whether a change was an improvement.

From instruction to system

A well-written instruction is where this starts. It is not what you operate. The difference between the two is not the wording — it is everything wrapped around it.

- **The instruction is versioned.** It lives as a file with a history, not as text in someone's head or in a chat window they scroll back through. You can see what changed, when, and why.
- **There is a regression test set.** A fixed set of real cases with known correct answers, re-run every time the instruction changes. Without this you cannot tell an improvement from a coincidence.
- **Deterministic checks wrap non-deterministic output.** Anything mechanically checkable is checked mechanically, not by reading: does the total cast, does the date fall in the period, does the reference resolve, is the value inside the expected range. The model proposes; ordinary code verifies. This is the single largest difference between something that works in a demonstration and something that works on a Tuesday in March.
- **Steps are orchestrated, not pasted.** In multi-step work, each step's output is validated before it becomes the next step's input, and there is a defined path for what happens when a step fails.
- **Changes are owned.** A named person approves changes to the instruction, and the change is logged. Instructions drift otherwise, and nobody notices until the output is wrong.
- **There is an audit trail.** Input reference, output, reviewer, decision, date — retained and retrievable.

Most small businesses can start with the first, second and last of these and capture most of the value. But it is worth knowing what the full shape looks like, because the gap between "we wrote a good prompt" and "we operate a workflow" is exactly this list — and it is where the reliability lives.

6. Data, tools and the Singapore layer

Platform-agnostic design means the workflow survives a change of model, application or vendor. Specify capabilities and interfaces, not brand names.

Capability map

CAPABILITY	WHAT TO SPECIFY	FALLBACK IF UNAVAILABLE
Generate	Draft, transform, explain, create variations	Reusable templates
Retrieve	Find relevant approved information	Curated knowledge base
Classify	Assign category, priority or route	Rules-based triage
Extract	Turn documents into structured fields	Standard form or manual capture
Analyse	Compare, summarise, identify patterns	Spreadsheet review with sampling
Orchestrate	Move work between people and systems	Checklist and calendar cadence

VENDOR-NEUTRAL REQUIREMENT

Document the workflow in terms of inputs, outputs, permissions, review gates and metrics. The tool will change. These operating requirements should not.

The four questions to ask before any data goes anywhere

1. **Where does it go, and who can see it?** Consumer-grade tools and business-grade tools differ mainly in the answer to this question, and the difference is not visible in the interface.
2. **Is it personal data?** Names, contact details, identification numbers, salary, health — anything about an identifiable individual.
3. **Is it a record you are required to keep?** If an output becomes part of your accounting or business records, it inherits every retention obligation those records carry.
4. **Would you be comfortable explaining this use to the person the data is about?** This is not a legal test. It is a reliable early-warning system.

Data handling rules

- **Classify before use** — public, internal, confidential, restricted. Decide once, for categories of information, so nobody has to decide in the moment.
- **Minimise** — remove names, identifiers and unrelated detail where the task does not need them. A supplier invoice can usually be coded without the contact person's mobile number.

- **Prefer approved source sets** over open-ended browsing for anything that informs an internal decision.
- **Keep a human-readable record** of source, transformation and approval.
- **Define retention** — what is stored, where, for how long, and who can access it.

The Singapore layer

Three regimes shape almost every AI decision in a Singapore business. None of them mentions AI. All of them apply to it.

Personal data (PDPA). If your workflow touches personal data, the ordinary obligations follow it into the tool: you may only use the data for purposes the individual was notified of and consented to; you must protect it; you must not retain it once the purpose has passed; and you remain responsible for it even when a third party processes it on your behalf. Overseas transfer carries its own condition — the recipient must be bound to a comparable standard of protection, and using a tool hosted outside Singapore is a transfer. Organisations must also appoint a data protection officer, and a data breach that is likely to cause significant harm, or that affects a significant number of individuals, must be notified to the Commission and, where relevant, to those individuals. Practically: decide, in advance and in writing, which categories of personal data may enter which tools, and make the safe option the easy one.

Records (Companies Act and IRAS). Singapore companies must keep accounting records and supporting documentation for five years, in a form that allows transactions to be explained and financial statements to be prepared. Where AI produces something that forms part of that trail — a coding decision, a reconciliation, a schedule — the output is a record like any other. It must be retrievable, legible and explicable years later. The question an inspector asks is not "did a machine do it?" but "can you show me how this figure came to be?"

Machine-readable reporting is arriving anyway. Singapore is progressively moving GST-registered businesses onto e-invoicing through InvoiceNow, with invoice data transmitted to IRAS. The requirement began with newly incorporated voluntary registrants in late 2025, extended to new voluntary registrants from April 2026, and expands in waves from 2028 to cover all GST-registered businesses by April 2031. The strategic point for an owner is simple: your invoice data is becoming structured and machine-readable to the regulator on a published timetable. It is worth making it structured and machine-readable to *you* first — the same discipline that satisfies the mandate is what makes your own automation possible. Check the current position with IRAS, as timelines are periodically updated.

A NOTE ON PROFESSIONAL WORK

AI can prepare almost anything in the finance and compliance chain: schedules, reconciliations, computations, draft filings, draft statements. It cannot be the accountable party. Statutory filings are made by the licensed people responsible for them; audit opinions are signed by a registered public accountant; directors remain responsible for the accounts they approve. Any arrangement that blurs this is a governance problem regardless of how well the technology performs.

7. Governance, risk and the accountability line

Governance should make safe action *easier*, not slower. Use light controls for low-risk work and stronger controls as impact, sensitivity or irreversibility increase. A governance regime that treats drafting an internal summary the same as approving a payment will be ignored, and deserves to be.

Four risk tiers

TIER	TYPICAL WORK	REQUIRED CONTROLS
1 — Assistive	Drafting, summarising, formatting, internal explanation	Approved inputs; user reviews before use; no autonomous external action
2 — Operational	Routing, extraction, classification, internal recommendations	Test set; quality threshold; defined exception path; periodic audit sample
3 — Sensitive	People, pricing, eligibility, customer commitments, regulated or confidential data	Named accountable owner; restricted access; documented approval on every instance; recurring review
4 — Prohibited	Irreversible, deceptive, discriminatory or unreviewable decisions	Do not deploy. Redesign or escalate.

Most small businesses can operate almost entirely in Tiers 1 and 2 for a long time and capture most of the available value. The pressure to move up the tiers usually comes from enthusiasm rather than from evidence.

The accountability line

Some decisions are not a question of how good the technology is. They are a question of who answers for the outcome. In a Singapore business, a machine should never be the last step before any of these:

- **A payment leaving the business**, or a change to payment details
- **A statutory filing** — ACRA, IRAS, CPF, or any regulator
- **A signature** on financial statements, a director's statement, or a declaration
- **A commitment to a customer** on price, scope, delivery or liability
- **A decision about a person** — hiring, dismissal, discipline, remuneration
- **A response to a regulator, auditor or lawyer**
- **A legal or tax position**, as distinct from the preparation supporting one
- **Any decision where nobody could tell, afterwards, that it was wrong**

The list is short and it is not a technology limitation — it is a design principle. Preparation can be delegated almost without limit. Accountability cannot be delegated at all.

Minimum governance checklist

- ☐ Named business owner and workflow owner
- ☐ Purpose and intended users documented
- ☐ Data classification and access rules recorded
- ☐ Known failure modes and mitigations listed
- ☐ Human review and escalation criteria defined
- ☐ Test examples include edge cases and deliberately awkward inputs
- ☐ Monitoring metric and review cadence agreed
- ☐ Change log maintained for instructions, rules and source material

RED FLAGS

A high-impact decision is being made without review · the source of an answer is unknown · users cannot tell whether content was AI-assisted · personal or confidential data is being copied without a clear need · nobody owns the outcome · the only person who understands the workflow is the person who built it.

8. Adoption, measurement and improvement

Adoption is a management problem wearing a technology costume. People use AI when it reduces friction, fits their existing rhythm, and does not expose them to blame.

Adoption design

- Give the workflow a visible owner and a small group of early users.
- Train on **the task**, not on abstract AI concepts. Nobody needs to understand a model to use a checklist.
- Provide examples of a good input, a good output and a good review.
- Make the safe path the easy path — templates, defaults, pre-approved sources.
- **Celebrate useful corrections.** If people believe that finding a fault reflects badly on them or on the project, they will stop reporting faults, and you will lose your only early-warning system.
- Be explicit that this is not a headcount exercise if it is not one. If people suspect otherwise, adoption data becomes meaningless.

Measurement stack

LEVEL	MEASURE	EXAMPLES
Outcome	Did the business result improve?	Cycle time, cash collection, conversion, satisfaction, error rate
Workflow	Did the process improve?	Completion time, handoffs, rework, adoption rate
Quality	Was the output fit for purpose?	Accuracy sample, completeness, policy adherence
Risk	Did exposure stay within bounds?	Escalations, incidents, sensitive-data events
Learning	Is the system getting better?	Correction themes, instruction changes, test-set score

The one number worth defending: **the edit rate**. What proportion of outputs does the reviewer change before use, and are the changes shrinking? A falling edit rate with stable quality is the clearest evidence that a workflow is maturing. A falling edit rate with *unexamined* quality means your reviewer has stopped reading, which is worse than where you started.

REVIEW RHYTHM

Weekly during the pilot: review ten representative cases. Monthly after adoption: review metrics, exceptions and changes. Quarterly: confirm the use case still deserves its risk and its maintenance cost. Workflows should be allowed to die.

9. Eight ways this goes wrong

Patterns we see repeatedly. None involves the technology failing.

- 1. Shadow use.** Staff paste customer data, contracts or payroll files into whatever consumer tool they have on their phone, because the official option is slow or does not exist. This is the most common data exposure in small businesses today, and prohibition alone does not solve it. Provide an approved option that is genuinely easier than the unapproved one, then set the rule.
- 2. The confident wrong answer.** Output that is fluent, plausible, correctly formatted and factually wrong. Fluency reads as competence. The defence is structural: require sources, require uncertainty to be stated, and keep test cases with known answers.
- 3. Review theatre.** A human is nominally in the loop but approves everything in seconds. This is worse than no review, because it manufactures a record of oversight that did not happen. If review is real, some things must get sent back — if nothing ever is, the control is decorative.
- 4. The one-person system.** Everything runs through the enthusiast who built it. They go on leave and the workflow stops. Anything that matters needs documentation and a second operator.
- 5. Automating a broken process.** The process was badly designed; now it is badly designed and fast. Map the work before you automate it, and be willing to conclude that the fix is not AI.
- 6. Scope creep without re-assessment.** A workflow approved for internal drafting quietly starts sending things to customers. The risk tier changed and nobody re-ran the assessment. Any change to who sees the output is a new decision.
- 7. Measuring activity instead of outcome.** Usage statistics rise; nothing about the business changes. Volume is not value. Go back to the baseline you wrote down.
- 8. No off switch.** Nobody defined what would make you stop. Every workflow needs a stop condition agreed in advance, when everyone is calm — because it will need to be used on a day when nobody is.

10. Practical checklists

Short by design. Add your own controls only where they reduce a real risk.

Before a pilot

- ☐ Outcome and baseline are written down
- ☐ Workflow owner is named
- ☐ Scope is narrow enough to test
- ☐ Inputs are approved and classified
- ☐ Human review is explicit
- ☐ Success and stop conditions are defined
- ☐ Test cases include normal and edge examples

Before releasing any output

- ☐ Every factual claim is supported by an approved source
- ☐ The output is complete for its intended use
- ☐ Tone is appropriate for the audience
- ☐ Assumptions and uncertainty are visible
- ☐ The right person has reviewed it
- ☐ Restricted information has been removed

Before letting a workflow touch anything external

- ☐ The risk tier has been re-assessed for the new audience
- ☐ A named person approves each instance, or a documented threshold governs which need approval
- ☐ There is a way to retract or correct
- ☐ The recipient can reach a human easily

After an incident or near miss

- ☐ Pause the affected workflow if harm could recur
- ☐ Preserve the input, output and decision record
- ☐ Describe what happened without assigning blame
- ☐ Identify the failed control, not only the bad answer
- ☐ Add a test case or a guardrail
- ☐ Communicate the change to affected users
- ☐ Re-enable only after the owner signs off
- ☐ Check separately whether any notification obligation was triggered

11. A 30-day implementation roadmap

Designed for a small organisation with limited capacity. Each week ends with evidence and a decision. Keep the team small: one sponsor, one workflow owner, two to five users, one person responsible for measurement.

DAYS	FOCUS	ACTIONS	EXIT EVIDENCE
1-3	Align	Name the sponsor; choose the outcome; set guardrails; select one workflow	One-page Success Contract
4-7	Observe	Map the work; collect examples; baseline time and quality; list failure modes	Workflow map + test set
8-14	Design	Draft the process; define review gates; create templates; train users	Working pilot + review checklist
15-21	Run	Use on low-risk cases; sample outputs; log corrections; fix friction	10-20 reviewed cases + findings
22-26	Evaluate	Compare to baseline; review risks; interview users; decide adopt / revise / stop	Pilot scorecard + decision
27-30	Institutionalise	Document the procedure; assign the owner; set the cadence; choose the next opportunity	Published procedure + 60-day backlog

Pilot scorecard

METRIC	BASELINE	PILOT	TARGET	NOTES
Cycle time				
Quality / error rate				
Edit rate on review				
Adoption / usage				
Review effort				
Incidents / escalations				

THE DECISION AT DAY 30

Adopt when the outcome improved, quality is acceptable, the controls are workable and the owner can maintain it. **Revise** when the value is visible but the process or the data is not ready. **Stop** when the risk or the maintenance cost outweighs the benefit. Stopping is a successful outcome — you bought that information cheaply.

Appendix A — The readiness scorecard

Score each statement 0 (not true), 1 (partly true) or 2 (true). Be honest; the score is only useful if it is.

#	STATEMENT	SCORE
Value		
1	We can name one workflow where a delay or an error costs us something we can measure.	
2	We know roughly how long that workflow takes today.	
3	Someone senior would notice if it improved.	
Readiness		
4	The inputs to that workflow are in a consistent, findable form.	
5	One named person owns the workflow end to end.	
6	The steps are stable enough that we could describe them to a new joiner.	
Risk and data		
7	We have decided which categories of information may and may not go into external tools.	
8	We know which of our data is personal data, and who is responsible for it.	
9	Nothing in the intended workflow ends in an irreversible action without a person.	
Governance and learning		
10	There is a named reviewer, and they have the time to actually review.	
11	We have written down what would make us stop.	
12	We have a handful of real examples with known correct answers to test against.	
Total (out of 24)		

BAND	SCORE	WHAT IT MEANS	DO THIS NEXT
Not yet	0–8	The groundwork isn't there. AI would amplify existing disorder.	Fix one process and one data question first. Chapter 2, then Chapter 6.
Prepare	9–15	A real opportunity exists but the controls or inputs aren't ready.	Write the Success Contract; sort out the input boundary; then pilot.
Pilot ready	16–20	You can start safely on a bounded workflow.	Run the 30-day roadmap in Chapter 11, unmodified.
Scale ready	21–24	You have the discipline to run several workflows.	Move from single pilots to a prioritised portfolio using Chapter 3.

Appendix B — Starter task specifications

The specifications below are written as prompts because that is the portable form — you can paste one into any capable system and get useful work back today. Treat them as starting points, not artefacts. In an operating workflow each of these carries a version number, a named owner, a set of test cases with known answers, and a change log; the text is the smallest part of it. Replace the placeholders with your own language, approved sources and review standards.

1. Executive brief

Create a one-page brief for [audience] using only the supplied materials. Include: the decision needed, context, key facts, options, trade-offs, a recommendation, risks and next actions. Separate facts from interpretation. Cite the source section for each material claim. Flag gaps instead of filling them with assumptions.

2. Meeting to action register

Turn these notes into an action register. Extract decisions, open questions, actions, owners, due dates, dependencies and risks. Do not invent owners or dates. If a field is missing, mark it "Not specified." Return a concise table followed by the three most important unresolved questions.

3. Customer request triage

Classify each request into one category from [categories]. Provide priority, reason code, suggested next step and your confidence. Use only the routing rules below. If the request involves [sensitive cases], route it to a human and explain why. Never promise an outcome or state policy that is not in the rules.

4. Supplier invoice extraction

From the attached document, extract: supplier name, invoice number, invoice date, currency, net amount, tax amount, total, payment terms and line-item descriptions. Return as a table. If a field is absent or illegible, return "Not found" — do not infer it. List separately anything that looks inconsistent: totals that do not add up, a date outside the expected period, or a supplier not on the approved list.

5. Receivables follow-up draft

Draft a follow-up message for the overdue invoice below, matched to reminder stage [1/2/3]. Use a professional, non-accusatory tone. State the invoice reference, amount, original due date and one clear next step. Do not mention legal action, credit holds or interest unless explicitly instructed. If the account has an open dispute, an unapplied credit, or a payment arrangement, do not draft — return "Hold: human review required" and explain why.

6. Variance explanation

Compare this period to the prior period and to budget. For every variance above [threshold], explain the movement in plain language, citing the specific transactions or accounts that caused it. Distinguish clearly between (a) what the data shows, (b) what you infer, and (c) what you cannot determine from the information provided. Do not speculate about causes outside the data.

7. Quality reviewer

Review the draft against this rubric: accuracy, completeness, source support, clarity, tone, privacy and actionability. Return pass or revise for each criterion, quote the smallest problematic excerpt, explain the issue and propose a correction. Do not rewrite the whole draft unless asked.

8. Procedure builder

Convert the approved workflow notes into a standard operating procedure with purpose, scope, roles, prerequisites, steps, decision points, exception handling, quality checks, records and review cadence. Keep every step observable and testable. Mark any unresolved ambiguity as an open item rather than resolving it yourself.

Appendix C — The Success Contract

One page. Written before anything is built. Signed by the sponsor and the workflow owner.

- Outcome to improve: _____
- Workflow and owner: _____
- Pilot scope and dates: _____
- Baseline measure: _____
- Target measure: _____
- Human review and approval: _____
- Inputs allowed / inputs excluded: _____
- What must never happen: _____
- Stop condition: _____
- Review cadence and decision date: _____

About this playbook

Nebis is a Singapore accounting and professional services firm built AI-native from the start. AI systems perform the preparation — extraction, reconciliation, scheduling, drafting — while qualified professionals direct the work, review it, and take personal responsibility for every regulated outcome. Audit opinions are signed by the firm's registered public accountant; statutory filings are made by the licensed people responsible for them. AI prepares; people decide, sign and file.

This playbook contains general guidance, not professional advice, and does not take account of your particular circumstances. Regulatory requirements change; verify current obligations with the relevant authority or your own adviser before acting.

Nebis.AI · Singapore

Edition A — for owners, founders and finance leads. A companion edition for accounting and audit practice owners is also available.