STANDARD OPERATING PROCEDURE

Change by Design · Practical AI. Practiced Hand.

CxD SOP-004 — AI discovery and delivery master

The procedure Change by Design runs an AI engagement from — opportunity discovery to scaled adoption. Published unabridged from our operating manual: every phase, question bank, rubric, and stop condition we actually use.

DOCUMENT	REVISION	DATE	OWNER	STATUS
CxD-SOP-004 · AI discovery and delivery master	2.15	2026-07-06	Dave Taylor	Live

0. Purpose, scope, outputs

Purpose. A repeatable procedure for prioritising the best AI use cases in any business: find the issues, choose the ones AI resolves, design the intervention, prove it against a baseline, scale what proves.

Run this when: starting any client engagement (it is weeks 1–2 of the 90-day Sprint and the onboarding month of Advisory) or re-scoping a stalled programme.

Outputs, in order produced:

1. Operating-model read + hunting-grounds shortlist (Phase A)
2. Company-specific questionnaires, one per interviewee (Phase B, step 1)
3. Interview notes + collected artefacts (Phase B)
4. Touchpoint logs from observation sessions (Phase C)
5. Tool & exposure inventory (Phase D)
6. Scored use-case pipeline; 2–3 pilots selected (Phase E)
7. Baseline sheet per pilot, signed by the workflow owner (Phase F)
8. Readiness-gate sign-off (Phase G)
9. Instrumented pilot + checkpoint decision (Phase H)
10. Rollout package per proven pilot (Phase I)
11. Vendor decision memos as tooling decisions occur (Phase J)
12. Weekly / monthly / quarterly reporting per cadence (Phase K)

Sector adaptation. The procedure and question modules below are generic — written around "units of work" (a tax return in a practice; an order, quote, or changeover in a plant). Section 14 maps the vocabulary across services, manufacturing, and distribution/trade. Every

engagement gets company-specific questionnaires derived per Phase B step 1; the modules here are the master coverage checklist those questionnaires must satisfy.

1. Operating rules

These settle disagreements. Everything downstream assumes them.

1. **Adoption over deployment.** A deployed tool nobody uses is a cost. Plan coalition, champions, measurement, and leadership air cover before planning software.
2. **Measure against their baseline, not ours.** Every claim gets a number built from the client's own data, captured before anything changes. External benchmarks are placeholders only, and get labelled as such.
3. **Value first, tools last.** Start from where money, hours, and errors concentrate. Never start from a tool. If the cheapest reliable fix is a template or deleting a step, that is the recommendation.
4. **Restrict before scale.** No pilot touches a messy estate. Grounded tools see allow-listed, permission-checked sources only, until each new source passes the same checks.
5. **Humans in the loop wherever an action has consequences.** Moving money, external comms in the firm's name, deletions, decisions about a person → human checkpoint **before** the action completes. Low-stakes drafting and lookup run free — over-gating trains click-through.
6. **Capacity, never headcount.** Report saved hours as capacity, margin, throughput, or client time. Headcount framing kills adoption; do not use it, in writing or in the room.

2. Phase A — Read the operating model

Goal: pick 2–3 hunting grounds (functions where a saved hour or avoided error is worth the most) before interviewing anyone below leadership.

Step A1 — Request the inputs. Before the first interview, request:

- [ ] Org chart with headcount by function and site
- [ ] Revenue model description: price list / fee structure / unit pricing; split of fixed-fee vs time-billed vs unit-margin revenue
- [ ] Volumes: units of work per year, by type (jobs, quotes, orders, tickets, returns)
- [ ] Loaded cost rates by grade or role (or salary bands + employer-cost % to compute them)
- [ ] Systems list: systems of record, line-of-business tools, comms stack
- [ ] Any prior automation/AI attempts and what happened to them
- [ ] Existing management reports/dashboards leadership already reads

Step A2 — Answer three questions from the inputs (confirm in Phase B, never assume):

1. **Where does the money come from, on what terms?** Decides what a saved hour is worth: fixed-fee → margin + capacity; time-billed → capacity only unless repriced; unit-margin → throughput and cost-per-unit.

2. **What is the binding constraint?** Hiring, quote turnaround, plant capacity, rework, client churn from slow turnaround. AI aimed at the constraint moves the business; aimed anywhere else it moves a chart.
3. **Where do hours and errors concentrate?** High-volume repeatable work; per-unit savings compound there.

Step A3 — Write the hunting-grounds shortlist. 2–3 functions, each recorded as:

- Function / workflow family
- Why it's a hunting ground (volume × value logic, one paragraph)
- What a saved hour is worth here and where it lands
- Known unknowns to resolve in Phase B

Rule: two or three hunting grounds. Not ten. If a fourth looks compelling, it queues.

3. Phase B — Discovery interviews

Goal: complete coverage of what each level of the organisation knows, with numbers and artefacts, not impressions.

B1 — Derive the company-specific questionnaires

The modules in §4 are the master coverage checklist. For each engagement:

1. Map the module roles to the company's actual titles (a "workflow owner" might be a branch manager, a practice lead, a plant manager, a head of estimating). One questionnaire per named person.
2. Rewrite each question in the company's vocabulary — their systems, their unit of work, their job titles.
3. Add company-specific themes surfaced by Phase A (e.g. an acquisition being absorbed, a new ERP mid-rollout, seasonality).
4. Check coverage: every theme in the relevant §4 module appears in the derived questionnaire, or its omission is a deliberate, noted decision. **A theme silently missing = the procedure failed.**
5. Save questionnaires with the engagement records (client folder), one file per interviewee.

B2 — Sequence and logistics

- Order: leadership (CEO/MD, then CFO) → workflow owners per hunting ground → doers → exception handlers/QA → Technology/IS → HR/L&D. Leadership first: their answers scope everything below.
- One role per session. 45–60 minutes. Observation (Phase C) is booked separately — never merged into an interview.
- Run the workflow-owner module once **per hunting-ground function** (a sales-ops lead, a head of estimating, and a payroll manager each get their own pass).

- Record numbers verbatim. If an interviewee estimates, mark it `est.` and get the source for the real figure.
- Two questions get asked of every senior interviewee, always: **"What prompted this now?"** and **"How will you judge whether this worked at 12 months?"**
- The amnesty question (4.4 Q14) is asked of **every interviewee at every level** — leadership included. Shadow tools run at every grade, and hearing the top answer it honestly is what makes the amnesty credible below.

B3 — Rules of engagement

- **Interview the doer, not just the manager.** Managers describe the process as designed; doers describe it as it runs. The delta is usually the opportunity.
- **Watch the work, don't survey it** — that's Phase C; use interviews to select whom to watch.
- **Amnesty framing for tool questions.** "What do you actually use to get work done?" asked as curiosity. An audit tone produces the sanitised answer. Asked at every level, not just doers.
- **Name sceptics and champions as you go.** Keep a running list, by name, from the first session.
- **Collect artefacts, not just answers.** Every module ends with a "walk away with" list. Leave each session with the items or a named owner + date for them.

4. Interview modules (master question banks)

Format per module: **Purpose → Questions by theme → Probes → Listen for → Red flags → Walk away with.** Questions are generic by design; B1 specialises them per company.

4.1 CEO / MD (or GM / site director)

Purpose: mandate, decision rights, constraint, risk appetite, air cover. This interview scopes every other one.

Mandate & trigger

1. What prompted this initiative/role now — what changed?
2. What have you already tried with AI or automation? What happened to it?
3. How will you judge whether this worked at 12 months? What number moves?
4. What does failure look like? What outcome would make you stop funding this?
5. Who wanted this — you, the board, a competitor scare, a customer? Who is the real sponsor?

Strategy & constraint

6. What's the growth plan for the next 2–3 years, and what breaks first if volume doubles?
7. What is the binding constraint today — hiring, capacity, turnaround, quality, cash?
8. Which parts of the business make the money, and which parts absorb the hours?
9. What can competitors do that you can't, and the reverse?

Decision rights & governance

10. What's a management decision versus a board decision here? Where does spend sign-off sit, at what thresholds?
11. Who can veto this programme, formally or in practice?
12. How are decisions actually made day to day — committee, consensus, one desk?

Risk & culture

13. What's the worst thing AI could do to this business? What must never happen?
14. How much appetite is there for visible experiments that might fail?
15. Who will resist this, and why? Who will champion it without being asked?

Air cover

16. How much of your own time will this get? Will you use the tools yourself, visibly?
17. Who announces this to the organisation, and how?

Probes: "Tell me about the last change programme here — what stuck, what didn't, why." · "If I gave you back 10,000 hours next year, where would you spend them?" **Listen for:** the gap between the stated trigger and the real one; whether the 12-month number exists at all; who they name unprompted. **Red flags:** no 12-month number ("we'll know it when we see it"); sponsor is a delegate, not the person in the room; headcount-reduction framing from the top. **Walk away with:** the 12-month success definition (or an agreement that defining it is deliverable one); decision/spend thresholds; sponsor time commitment; first names for the sceptic/champion list.

4.2 CFO / finance lead

Purpose: revenue mechanics, cost concentration, volumes, and how business cases get judged. This interview parameterises the value model.

Revenue mechanics

1. Walk me through how revenue is earned: what's fixed-fee, what's time-billed, what's unit-margin, what's project? Rough percentages of each.
2. Where does margin concentrate — which service lines, products, customer types?
3. If a process gets faster, where does the saved hour land in the P&L? Margin, capacity, or lost billings?
4. What's repriced / how often? Could time-billed work be repriced if it got materially cheaper to deliver?

Cost & volumes

5. What are the loaded cost rates by grade/role (salary + employer costs ÷ productive hours)? If not computed, what are the inputs?

6. What are the volumes: units of work per year by type? Which is the biggest line?
7. Where is overtime, temp, or contractor spend concentrated? What drives it?
8. What does an error or a rework cycle cost when it happens? Any figures on write-offs, credits, penalties, scrap?

Measurement & approval

9. What numbers does the board already see monthly? Which report would this programme's results live in?
10. How do business cases get approved here — payback threshold, hurdle rate, gut feel? What's the smallest case that still needs formal approval?
11. What's the current spend on software/AI licences? Who owns that budget? Any shelfware you already suspect?
12. What would make you kill a programme like this mid-year?

Probes: "Show me the management pack you got last month." · "Pick your highest-volume unit of work — what does one unit cost to deliver, all-in?" **Listen for:** whether loaded rates exist or must be built; the real approval path vs the official one; scepticism to pre-empt in the business case. **Red flags:** no unit costing anywhere; licence spend unknown or unowned; payback expectations under 6 months for anything. **Walk away with:** volumes by unit of work; loaded rates (or inputs to compute them); revenue-split percentages; the approval threshold and format the business case must fit.

4.3 Workflow owner (run once per hunting-ground function)

Purpose: the process as designed vs as run, with stage-level times, queues, error rates, and prior attempts. This interview produces pilot candidates.

Process walk

1. Walk me through one unit of work end to end: every stage, every handoff, every system it touches. (Whiteboard or screen-share; capture the stages.)
2. Which stages are done by whom, at what grade/cost?
3. Where does a unit wait? For whom or what? For how long, typically?
4. Which handoffs lose information — where do people re-explain, re-key, or chase?

Volumes & timing

5. How many units per week/month/year? Seasonality, peaks, deadlines?
6. Time per unit, by stage — measured, or gut feel? Where's the widest variance between fast and slow staff?
7. What's the turnaround promise to the customer (internal or external), and how often is it missed?

Errors & rework

8. What comes back? Top three comeback types, roughly how often?
9. What does a comeback cost — time, money, relationship?
10. Which stage produces the most errors, and is that stage understaffed, undertrained, or badly tooled?

People & tools

11. Who are your strongest and weakest performers on this workflow, and what does the strongest do differently?
12. What tools does the team use — sanctioned and otherwise? What was tried and abandoned, and why?
13. What have you already tried to fix here? What happened?
14. If you could remove one 30-minute chunk from every unit, which would it be?
15. Who on your team would pilot something new well? Who would resist it loudest?

Probes: "Show me a finished unit — walk me through what it took." · "What did the last new starter struggle with longest?" (That's where the tribal knowledge is.) **Listen for:** stages the owner can't describe precisely (they've drifted from the floor); variance between performers (a template/knowledge gap, often automatable); "we tried that" stories — the failure mode usually names the readiness gap. **Red flags:** no time-per-unit sense at all; error rate unknown; the owner wants a tool named before the problem is described. **Walk away with:** the stage map with times (or a plan to sample them in Phase C/F); volume figures; top comeback types; 2–3 named candidate touchpoints; names for the pilot cohort and the sceptic list.

4.4 The doers (front-line staff who produce the unit of work)

Purpose: where the hours actually go — the repetitions, lookups, re-keys, and workarounds that never appear in the process map.

The actual day

1. Walk me through yesterday, start to finish. What ate the most time?
2. What do you do more than five times a week that feels the same every time?
3. What's the most annoying 30 minutes of your day or week?
4. What part of the job do you wish you could hand to someone else?

Repetition & drafting

5. What do you write repeatedly — letters, summaries, notes, reports? From what starting point (blank page, old example, template)?
6. Where do you keep your own cheat sheets, snippets, or side-spreadsheets? (Ask to see them.)

7. What's the last thing you copy-pasted between two systems? How often does that happen?

Lookup & knowledge

8. What do you search for most? Where? How long does it take to find, and how do you know what you found is current?
9. When you're stuck, who do you ask? What do you ask them? What happens when they're on leave?
10. What question do new starters ask you constantly?

Waiting & interruptions

11. What do you wait on before you can finish a unit? How long?
12. What interrupts you most? What can't you finish in one sitting, and why?

Quality & change

13. What gets bounced back to you, and by whom? What would you fix first if you owned the process?
14. Amnesty question (asked at every level, per B2 — canonical phrasing lives here): what tools do you actually use to get work done — including personal ChatGPT, browser extensions, home-made macros? (Curiosity tone. No consequences.)
15. What was the last new tool that actually stuck for you? What was the last one that didn't, and why?

Probes: "Show me." (For any repetition claim — watch one live.) · "How long did that take last time? Pull one up." **Listen for:** the workaround they mention casually — it's usually the finding; the same 20–30-minute chunk appearing in multiple answers; shadow AI already in use (a map of demand). **Red flags:** answers that exactly match the manager's process description (rehearsed or supervised session — re-run one-to-one); fear in the amnesty answer (adoption risk to log). **Walk away with:** a list of repeated touchpoints with rough per-unit minutes; copies/screenshots of cheat sheets and side-spreadsheets; the shadow-AI list for Phase D; 1–2 names of natural pilot users.

4.5 Exception handlers / QA / reviewers

Purpose: where errors surface, what they cost, and which controls are real. Reviewers see the whole system's failure modes.

1. What lands on your desk? Top three categories, rough weekly volume of each.
2. For each: what's the root cause, and at which stage did it enter?
3. What does one instance cost to fix — your time, others' time, customer impact?
4. What was the worst incident in the last year? What nearly happened?
5. How much of your review time is spent on units that turn out fine? Could anything pre-sort those?

6. Which checks in the process actually catch things, and which are box-ticking that everyone clicks through?
7. Which teams, sites, or stages generate disproportionate errors? Why, in your view?
8. What do you fix silently that nobody upstream ever hears about?
9. If error volume dropped by half, what would you do with the recovered time?

Probes: "Show me the last three items you handled — walk me through each." **Listen for:** silent fixes (invisible cost, strong pilot candidate); checks that are theatre (governance debt); a stage repeatedly named across interviews. **Red flags:** no records of comebacks (the error rate is unmeasurable — baseline will need building); reviewer as the single point of failure. **Walk away with:** comeback categories with frequency + cost-per-instance; the incident list; which controls are real; error-rate data or a plan to sample it.

4.6 Technology / IT / IS / data owners

Purpose: what's feasible, what's permitted, where data lives, and what gets vetoed. Run this module early, in discovery — every veto and constraint gets surfaced and designed around here; one that first appears at go-live means this module failed.

Estate & data

1. What are the systems of record? For each hunting-ground workflow: where does its data actually live?
2. Where does knowledge live — document stores, shared drives, legacy servers, email, people's heads? What's the messiest part of the estate?
3. What are the integration surfaces: APIs, exports, connectors? What's genuinely reachable vs walled off?
4. How healthy is data quality in the systems that matter — duplicates, stale versions, free-text fields doing structured work?

Permissions & identity

5. How healthy is the permission model? Org-wide shares, broken inheritance, anonymous links?
6. Stale accounts — leavers, contractors, acquired-company migrations?
7. If a grounded tool indexed everything a normal staff login can technically reach today, what could surface that shouldn't?

Current AI posture

8. What AI is deployed, trialled, or blocked today? Licences owned vs used?
9. Is there an AI policy? Who owns it? What does it actually prohibit?
10. What DLP, labelling, or monitoring exists? What alerts does anyone actually read?

Constraints & capacity

11. What would you veto outright, and why? What are the security/compliance non-negotiables?
12. What's the change-control path for a new tool or integration — steps, sign-offs, realistic elapsed time?
13. Who on your team could support pilots, at what fraction of their time? Which vendors are already in the building?
14. For the systems that matter: connector into the index, scheduled export, or keep it out and build a separate retrieval layer — what do the APIs actually allow?

Probes: "Show me the oversharing report, if one has ever been run." · "What happened with the last third-party integration that went badly?" **Listen for:** the veto criteria (design pilots inside them); the delta between official policy and observed practice; whether IT will be an ally or a bottleneck — plan support either way. **Red flags:** nobody owns permissions; no test/sandbox environment; change-control latency measured in quarters. **Walk away with:** systems + integration map for each hunting ground; permission-health read (or a date for the audit); the veto list; named IS contact for the readiness gate.

4.7 HR / L&D

Purpose: the change machinery — how training lands, who champions, what the workforce sensitivities are.

1. How is training delivered today — format, cadence, per-role or generic? What works here and what gets ignored?
2. Tell me about the last rollout that stuck. And the last one that didn't. What was the difference?
3. Who adopted the last new tool first? (Names — these seed the champion list.) Who never did?
4. Who are the credible-middle people — respected, not gadget-chasers — whose endorsement moves their peers?
5. What are the workforce sensitivities around AI — job-loss fear, works council/union constraints, comms protocols I must respect?
6. What's the realistic training capacity — hours per person per quarter the business will actually release?
7. How is performance measured today? Would time-saved data at individual level be acceptable, or does measurement need aggregating?
8. What's the AI literacy baseline — has anyone been trained on anything AI yet? (EU AI Act Art. 4 literacy duty applies to deployers.)
9. Who should hear about pilot results first, and through what channel, so the grapevine works for us rather than against us?

Listen for: the real reason the failed rollout failed (it repeats unless designed against); measurement sensitivities that shape Phase F instrumentation. **Red flags:** training is "a portal"; champions = whoever is free; individual-level monitoring is a hard no that nobody

mentioned earlier. **Walk away with:** champion candidate names; training-capacity numbers; comms constraints; the literacy baseline.

5. Phase C — Observation (watch the work)

Interviews collect claims; observation collects facts.

- [ ] Pick 2–3 doers per hunting ground (from 4.3 Q15 / 4.4 findings) — include one strong and one average performer.
- [ ] 60–90 minutes at their desk/station while they do real work. Not a demo — their actual queue.
- [ ] Capture a **touchpoint log**: timestamp, task, system/tool in use, switches between systems, lookups (what + how long), re-keys, waits, workarounds.
- [ ] Do not coach, suggest, or react. Record.
- [ ] Close by asking about anything surprising: "You did X three times — is that typical?"

Output: touchpoint log per session; per-unit minutes for candidate touchpoints (these seed Phase F sampling).

6. Phase D — Tool & exposure inventory

One row per tool that touches the work. Sources, in order of completeness:

- [ ] Last 3 months of card statements + expense claims — flag every software line
- [ ] Connected-app / OAuth-grant export from the M365 or Google Workspace admin console
- [ ] Browser extensions across the team (sample if large)
- [ ] The amnesty ask, per 4.4 Q14
- [ ] IT's licence list (4.6 Q8) — reconcile against the above; the deltas are the shadow stack

Row fields: tool · who uses it · what data it sees · answer-only or can act · what it can reach (systems, inbox, bank) · reversibility of its actions · sanctioned? · notes.

Output: inventory sheet. It is simultaneously the risk register input (Phase G) and an opportunity map — where staff smuggle tools in, they are marking friction.

7. Phase E — Score and select

E1 — Use-case card (one per candidate)

FIELD	CONTENT
Name	Verb + workflow ("Draft accounts-prep narrative")
Problem	One paragraph, from interview/observation evidence

FIELD	CONTENT
Users	Roles + count
Unit of work + volume/yr	From CFO/owner numbers
Touchpoint minutes	Per-unit minutes claimed (interview) + observed (Phase C)
Value math	Volume × hours × loaded rate, or error rate × cost-per-error; inputs sourced
Data needed + where it lives	From 4.6
Answer or act	Classification + human-checkpoint need
Risk class	EU AI Act tier + data sensitivity
Readiness gaps	Estate/permission/ownership issues blocking it
Owner	Named workflow owner who wants it
Success metric + baseline source	What Phase F will capture
Stop conditions	Tripwires / checkpoint thresholds / caps for this pilot (types defined in Phase H)

E2 — Scoring rubric (anchored 1–5 per axis)

Value — annualised value of the friction, from the card's math:

- 1: < 0.25× programme cost · 2: 0.25–0.5× · 3: 0.5–1× · 4: 1–3× · 5: > 3×

How to read this scale: take the annualised value from the use-case card (volume × hours saved × loaded rate, or errors avoided × cost per error) and divide it by the total programme cost for the year — fees plus tooling. That multiple is the score. Example: the programme costs the client €40k and a use case is worth €60k a year → 60 ÷ 40 = 1.5×, which lands in the 1–3× band and scores 4. A use case scoring 5 pays for the whole programme more than three times over on its own; a 1 returns less than a quarter of what the programme costs.

Feasibility — can AI do this well today:

- 1: data unreachable or task is open judgment with expensive errors
- 2: data reachable with major integration work; task partly judgment
- 3: data reachable; task structured; human verification slower than doing the work
- 4: data reachable; task structured and checkable; verification fast
- 5: proven pattern (draft-plus-review, retrieval on curated sources); verification trivial

Readiness — is the business ready for this pilot to succeed:

- 1: estate messy AND permissions leaky AND no owner

- 2: two of the three
- 3: one of the three, fixable inside the pilot window
- 4: minor gaps; owner engaged
- 5: clean scoped sources, safe permissions, owner pulling for it

Decision rule: score all candidates. **Advance** = no axis ≤ 2 **and** total ≥ 11. **Park** everything else with a one-line reason (readiness gaps often clear in Phase G — parked ≠ dead). Cap live pilots at **2–3**. First pilot = highest readiness among advancers, not highest value — the first win buys the second.

Vocabulary bridge (for external documents): this triad compresses to "ROI × readiness" and expands to "impact / feasibility / ROI". Same scores.

8. Phase F — Baseline

No pilot starts without a signed baseline. Per selected pilot:

- [ ] **Time per unit:** system timestamps where they exist; otherwise a 10+ unit sample across ≥2 performers. Record method used.
- [ ] **Turnaround:** request-to-delivery elapsed time, same sample.
- [ ] **Error/rework rate:** from QA records (4.5) or a sampled review.
- [ ] **Current tool usage:** what's used today on this workflow, how often.
- [ ] **Answer-quality bank** (knowledge/retrieval use cases only): 30–50 real questions staff actually ask, with known-correct answers, sourced from doers + reviewers. Score the tool against it **before** the pilot. This doubles as the improvement instrument.
- [ ] Workflow owner signs the baseline sheet. This is the number results get measured against — agreed now, not renegotiated at the checkpoint.

Metric tiers (wired now, reported per Phase K):

TIER	METRICS	READS AS
Usage (leading)	who used it, frequency, thumbs-down patterns	is it being adopted?
Operations	time/unit, turnaround, accuracy vs bank, error rate	is it working?
Money (lagging)	value model with measured inputs, run cost, net	did it pay?

Usage leads: a pilot with good accuracy and falling usage is dying, and shows it weeks before money does.

9. Phase G — Readiness gate (governance)

Run before any grounded tool sees data. Pass criteria per phase; no pass, no pilot.

G1 — Discover & assess

- [ ] Knowledge map: where content lives, per pilot scope (from 4.6, verified)
- [ ] Content audit: volume, age, duplication (expect well over half redundant/obsolete/trivial — normal)
- [ ] Oversharing audit: org-wide links, anonymous links, broken inheritance → risk register
- [ ] Identity hygiene: stale accounts, orphaned groups, leaver access
- [ ] Crown jewels named in writing: what must never surface in an answer for the wrong person

G2 — Lock down

- [ ] Kill org-wide/anyone links surfaced by the audit; fix sharing defaults to specific-people
- [ ] Sensitivity labels applied (simple taxonomy), auto-labelling for client/staff-data patterns
- [ ] DLP rules on the sensitive flows, including AI prompts/responses where the platform supports it
- [ ] Stale accounts purged; MFA/conditional-access gaps closed

G3 — Clean & structure

- [ ] Golden sources built: one canonical, owned, dated version per knowledge type; copies killed
- [ ] Named human owner per source (unowned content rots back within a year)
- [ ] Allow-list defined: the curated sources — and nothing else — the tool may ground on

G4 — Red-team & verify

- [ ] As a normal user account, per role: ask the questions we're afraid of (salaries, named-client data, redundancy plans). Every leak found = an incident prevented. Fix, retest.
- [ ] Score answer quality against the Phase F question bank; record the pre-pilot score
- [ ] Standing rule adopted in writing: **no new source joins the allow-list until it passes G1–G4 checks**

Classification rules (applied on the use-case card, Phase E):

- **Act vs answer:** answer-only tools → standard review flow. Acting tools → human checkpoint before: money movement, external comms in the firm's name, deletion/overwrite, decisions about a person (GDPR Art. 22). Rank acting tools by blast radius: worst thing before anyone notices × reversibility.
- **EU AI Act tier** recorded per use case at intake (most internal draft-plus-review tools land minimal/limited; literacy duty applies regardless). Nothing gets designed that compliance must later kill.

10. Phase H — Pilot

Design checklist (all boxes before day 1):

- [ ] One workflow, one small named cohort — volunteers plus the named champion inside it

- [ ] Runs only on the G3 allow-list
- [ ] Instrumentation wired to the Phase F baseline from day one
- [ ] Playbook v1 written: role-specific, embedded in the templates/systems the cohort already uses — never a separate destination to remember
- [ ] Checkpoint date set; stop conditions (from the use-case card) restated in the pilot doc
- [ ] Weekly review booked: usage, accuracy vs bank, thumbs-down patterns, incidents. Failures = design input.

Stop conditions (written before day 1, on the pilot doc — three types):

1. **Tripwires** — incident-class, stop *immediately*, no meeting: data surfaced to the wrong person, a confidently wrong answer reaching a client/external party, any Phase G gate breach. Fixing and re-entering requires re-passing G4.
2. **Checkpoint thresholds** — measured on the checkpoint date: minimum usage (e.g. ≥ half the cohort weekly), accuracy vs the question bank's pre-pilot score, time-saved vs baseline. Set the numbers per pilot when the baseline is signed.
3. **Caps** — no decision-grade result by the checkpoint date, or spend beyond the pilot budget, forces the checkpoint decision even if metrics are ambiguous.

The point of writing these on day 1: a pilot that has never faced a real *no* makes every other number decorative. The first cleanly stopped pilot is what makes the measurement credible.

Checkpoint decision (on the date, against the signed baseline):

- **Scale** — metrics beat baseline and thresholds, usage stable/rising → Phase I
- **Fix & re-run** — value visible, specific fixable failure → one iteration, new checkpoint. (This outcome exists so stop conditions don't murder good ideas that need one more turn.)
- **Stop** — a threshold or cap tripped with no fixable cause → write up why, park the card, move to the next advancer

By the checkpoint the **champion presents the results, not CxD/the programme lead**. If it only works with the lead in the room, it hasn't worked.

11. Phase I — Adoption & scale

Champion selection: from the credible middle (respected, not gadget-chasers — names from 4.7 Q3–4 and 4.3 Q15). One per site/function in scope. Train and equip champions; champions train peers. Rollout must not depend on the programme lead's presence.

The adoption pattern (use it everywhere): reframe the threat honestly as what it does *for the person* (client time back, engineering instead of re-keying) → let champions sell peer-to-peer → let baseline numbers close the argument.

Rollout package per proven pilot (standardisation):

- [ ] Playbook (role-specific, final)
- [ ] Onboarding pack + training materials
- [ ] Metrics to watch + known failure modes
- [ ] Local-adaptation notes: what may vary per site, what must not

Site two should cost a fraction of site one; if it doesn't, standardisation happened too little or too late.

Usage watch: usage falling for two consecutive weekly reviews → treat as a design problem, intervene (playbook fix, retraining, workflow embed), never blame users.

Air cover asks (explicit, to the sponsor): leaders use the tools visibly; pilot cohort's learning time is defended; results get announced through the channel agreed with HR (4.7 Q9).

12. Phase J — Build vs buy

Check routes in this order, per use case:

1. **Configure what's already owned** (M365/Copilot-class capability in the existing stack). Data stays in the estate; IT already governs it.
2. **Buy a point tool** — only where a mature product clearly fits. Assess: integration surface, data terms (what leaves, who trains on it), total cost incl. licences nobody cancels, lock-in/exit path, vendor viability.
3. **Build thin custom** — only where the workflow is differentiating or nothing fits. Keep thin: retrieval layers, glue automations, narrow-scope agents. Custom must keep earning its maintenance.

Vendor decision memo (every tool bought or killed): decision · options considered · data terms · cost · exit path · decider + date. Memos accumulate into the audit trail that answers "why do we pay for this?"

Licence rule: free/basic tier broadly; paid seats for demonstrated heavy users; expand when usage numbers demand it. Never license ahead of adoption.

13. Phase K — Reporting cadence

CADENCE	AUDIENCE	CONTENTS
Weekly (during pilots)	pilot team + owner	usage, accuracy vs bank, failure patterns, incidents, fixes
Monthly	leadership	shipped/slipped, results vs committed metrics, next month's three priorities
Quarterly	board / steering group	results vs baseline, spend, risk/compliance status, decision memos issued, next quarter's plan

Benefits tracking **continues after go-live** — projected vs realised on every scaled case. Realised value is the only kind reported without a qualifier.

14. Sector mapping — services vs manufacturing vs distribution

The procedure is identical; the units, the waste, and the people change. Use this to specialise Phase A/B artefacts:

	PROFESSIONAL SERVICES	MANUFACTURING	DISTRIBUTION & TRADE
Unit of work	compliance job, return, client letter, engagement	order, quote, changeover, batch, NCR	order, delivery, pick, return, SKU line
What you count	hours/job, turnaround, fixed-fee margin, capacity (FTEs)	throughput, quote lead time, scrap/rework, OTIF, cost/unit	order accuracy, OTIF, stock turns, pick rate, cost/order
Where waste hides	first drafts, summaries, rule lookups, client letters, knowledge search	re-keying across ERP/MES/CRM, quote assembly, scheduling, quality documentation, floor tribal knowledge	order entry + re-keying, stock/availability queries, carrier chases, returns processing, price lookups
Who fills the module roles	practice/branch leads, service-line leads, preparers, reviewers	plant managers, planners, quality, maintenance, supply chain, estimators, operators	depot/branch managers, buyers, planners, order-desk staff, warehouse supervisors, returns handlers
What you baseline	time/job, turnaround, review/rework rate	cycle time, changeover, quote lead time, error/scrap rate	order-to-dispatch time, order-entry time, order accuracy, returns/claims rate
Where the saved hour lands	fixed-fee margin + client face-time	throughput + freed engineering/planning time	order-desk capacity + service level (more orders, same team)
Governance emphasis	client confidentiality; liability of a wrong professional answer; GDPR	IT/OT boundary; group policy/IS alignment; safety-adjacent systems stay human-gated	pricing + customer-data confidentiality; stock and price files stay human-gated; EDI/ERP integrity
The sector's nightmare	the confidently wrong answer in front of a client	anything that could stop the line or corrupt the schedule	a wrong price or false stock promise going out to customers at scale

Weight the readiness gate toward answer-quality + golden sources in services; toward the act-vs-answer split + IT/OT boundary in manufacturing; toward price/stock-file integrity and human gates on customer-facing pricing in distribution.

15. Engagement mapping

- **90-day Sprint** = Phases A–F compressed into weeks 1–2 (audit & select), G–H in the build weeks, I + K in the handover weeks.
- **Advisory** = the SOP run continuously: the weekly day works Phases B–E as a rolling pipeline; each month's shipped workflow runs F–H; memos (J) and the monthly/quarterly packs (K) accumulate.

- **In-house role** = the same phases mapped onto a 30/60/90: A–F ≈ days 0–30, G–H ≈ 31–60, I + K ≈ 61–90.

About this document. This is CxD's working SOP, published as we run it. If you want it run in your business — or want the version of it your team runs after we leave — that is what an engagement is. hello@changebydesign.com

---