

Management as Ontology

The canon · v3.4.0 · 2026-08-29 · Norman Wanto (drafted with Alfred) · release gate PASS

Name it, clock it, write it, guard it, garden it.

The map (from canon/REGISTER.yaml)

Every unit below is identified by NAME; epistemic stamps: settled = receipts cited, provisional = ratified, first receipt pending. Part numbers are reading order only.

The seven models

- The three layers P1
- The two loops P1.25
- The five verbs P2.5
- The two doors P2
- The three skills P2
- The four enforcement roles P5
- The ball and the court P5

The seven instruments

- The catechism P3
- The seven fitness tests P4
- The five presences P2.5
- The jobs admission standard P2
- The four grain tests P2
- The five dispositions P2.5
- The ten-word mantra P6

The 26 laws, seven families

Laws of the nouns

- The identity law settled P1
- One noun, one home settled P1
- The re-homing rules settled P1
- The ranking corollary settled P1

Laws of the spec

- The reification principle settled P1.25
- Spec and implementation settled P1.25
- The spine anatomy settled P2.5
- Types and bindings provisional P2.75

Laws of the doors

- Jobs-first admission settled P2
- The depth rule settled P2
- The split rule settled P2
- The grain law settled P2
- Construction grounds provisional P1.75

Laws of the running system

- Two kinds of processes provisional P2.5
- The induction law provisional P2.5
- Inspect means exceptions settled P2.5
- Deferrals carry flip conditions provisional P4

Laws of enforcement

- Enforce artifacts, never conversations settled P5
- The calcification law settled P5
- Derivation before delegation settled P5

Laws of authority

- The authority law settled P1.75
- The stage curve settled P1.75

Laws of the head

- The cognitive-fit law settled P4
- The qualifier corollary settled P4
- Structure the pipeline, never the ideation settled P7
- Killable by evidence settled P7

Management as Ontology

For Gotrade leads · from Norman, drafted with Alfred · v3.4.0 (2026-08-29)

the model behind how Norman structures work, made explicit so it can be taught, practiced, and examined. A half-hour reference read; the teachable kernel is Parts 1-3, and the skill comes from the drills, not the reading.

Part 0: Why this matters (the hook, and it is not hype)

Two facts, both checkable against your own quarter:

- 1. Unstructured work does not compound.** A growth team without an experiment register repeats experiments and evaporates learnings: the same budget buys less insight every quarter. A marketing team without standardized output artifacts cannot be reviewed, handed off, or accelerated: every deliverable is bespoke, every review starts from zero. Work that is not written into named structures is work the organization forgets.
- 2. Structure multiplies agent leverage.** Gotrade already runs an agent fleet. Agents can handle ad-hoc asks anywhere, but they can gate an intake, pre-brief a standup, lint a register, or author an artifact only where those nouns exist. The department with ontology inherits the fleet's compounding leverage; the department without it gets an assistant instead of an operating system, and the gap widens every month.

Product, market, and distribution grow the company. This is what makes every unit of that work accumulate instead of evaporate, and it is the on-ramp for AI into your function. That combination is the highest-EV internal move available to a lead at Gotrade today.

Part 1: The three layers

Every well-managed domain has the same three layers. When work feels messy, unnamed, and untrackable, one or more of these layers is missing.

Layer	What it is	Gym example	Marketing example
1. Ontology	The nouns: what exists, its fields, its legal states, its identity scheme (how instances are named), and the declared places where it lives	Set, Rep, RPE, "Barbell Deadlift", the Daily Plan Notebook	Campaign, Experiment, Brief, Learning; the campaign register; the launch checklist
2. SOP	The verbs: what may be done to the nouns, by whom, on what clock	"Every morning, check the Notebook"	"Every Monday 10:00, standup walks the register: exceptions first"
3. Data	Today's instances, written in the layer-1 vocabulary	"Deadlift 5x8 @ RPE 7-8"	"EXP-041: IDN lookalike audience, CAC hypothesis, verdict pending"

The test of a good stack: **the daily instruction becomes one line.** "Deadlift 5x8 @ RPE 7-8" is ten words only because layer 1 defined Set, Rep, and RPE. Without the layers, every instruction is a bespoke 40-minute meeting, unrepeatable and unverifiable. With them, management's daily cost collapses to writing rows, and the team executes without the manager present.

The same layers in a second domain, to see the pattern generalize. Training for a marathon:

Layer	Marathon training
1. Ontology	Session types (long run, tempo, intervals, recovery), pace zones (Z1-Z5), weekly mileage, the training plan as the declared place
2. SOP	"Check the plan every morning; long run every Sunday; log every session"
3. Data	"Today: 16km @ Z2" and tonight's logged actual

The marathon adds what the gym example hides: **the loop over time**. Enforcement is visible (a coach or Strava sees the missed Sunday run the same day: skipping is loud), and gardening has a name in this domain, periodization: the plan amends itself on evidence (an injury, a fitness test, a bad week), and a plan that is never amended is how runners get injured. Lifting teaches the nouns; the marathon teaches the operating loop.

For engineers this is instantly familiar: **layer 1 is the schema and models, layer 2 is the functions and their scheduler, layer 3 is the rows and logs at runtime**. Management is systems engineering where the runtime is people.

The identity law

No noun without identity (the law of layer 1, in the spirit of Quine's dictum). A noun is not ontologized until its instances carry a DECLARED identity scheme, and two schemes are legal: a MINTED token where instances must be cited across time, people, or surfaces ("we track experiments" is vocabulary; "EXP-041" is an ontology), or a NATURAL KEY where the date or slot already individuates ("Tuesday's session" needs no serial number; the timestamp IS the scheme, declared).

Where tokens are minted, three rules: the scheme is set at ontologize time; every instance mints at birth, at insert not at approval (identity precedes status, so even a dropped thing stays citable); and IDs are permanent, never reused, never renumbered, surviving soft-delete, because the code is the audit trail's spine. The quality bar is HUMAN-SPEAKABLE: machines keep UUIDs underneath; the management tier is the conversational identity, one short token that survives being said aloud in a standup.

Identity is what the rest of the machinery stands on: the one-line instruction runs on it (the ID carries the context), enforcement needs it (you cannot make skipping loud for a thing you cannot name), inspection needs it (dispositions attach to IDs), receipts need it ("done" plus the row code is a verifiable claim; "done" alone is an assertion), and gardening needs it (you can only merge or kill what you can cite).

Namespacing is scoped by travel: a code that never leaves its workspace needs no qualifier (the workspace IS the namespace); a code that crosses boundaries carries its origin, added at the boundary by whatever carries the citation: a rendering surface where one exists, the citer's own discipline where the habitat is a plain sheet (the DNS model: short name inside the zone, fully qualified across zones).

(SETTLED: codes-at-birth running estate-wide. Ratified 2026-08-26; natural-key clause and qualification generalization per review v4, findings 4-5.)

The ranking corollary

Position is a field, never an identity (the identity law's corollary, position versus identity). Where ordering carries meaning — a ranked principles list — rank is an edition-scoped attribute of a stably-NAMED instance: cite by name, qualify rank by edition ("#1, Aug-2026 edition"; bare rank is legal only

inside the edition), and a reorder becomes an auditable diff that forces the amend-versus-replace call a silent renumbering dodges. Where ordering is mere arrangement — this canon's Parts — position is a pointer, free to restructure because nothing load-bearing cites it. Either way the identity never renumbers.

(SETTLED: ratified 2026-08-29 from the Gotrade principles-reordering question; each half already lived in the field — the code egress grammar, edition-scoped ranks.)

One noun, one home

Every noun lives in exactly ONE home domain (the jurisdiction law). The identity law names an instance; this law places the noun. The home domain's spine declares it and its owner stewards it; every other domain uses it BY TYPED REFERENCE, never by copy — a duplicated entity table forks truth (two SaaS registers cannot answer "what do we run"; the delta between them is invisible exactly where it matters). The split that makes sharing safe: the home owns the ENTITY — existence, identity, the universal fields, kept deliberately thin — while each referencing domain keeps its OWN FACTS about the entity in its own register, keyed by the home identity (finance's costs and the access owner's seats never share a table; both reference one SaaS row, and the reconciliation between them becomes a computable diff instead of two drifting spreadsheets).

Ownership: MINTED by whoever needs the noun first (the pain door, no committee), OWNED by the domain that best WITNESSES the entity's lifecycle — births especially — and bears the failure cost of an incomplete set; when first-filer and best-witness diverge, that is a jurisdiction dispute, ruled where grain disputes are ruled, and proposals stay open to every domain regardless (the home ratifies, and proposing must stay cheaper than forking).

DECISIONS obey the same law: a ruling lives in the spine of the domain holding jurisdiction over it (domain-scoped sequences, qualified at egress like every code); a ruling spanning domains belongs to the META-domain, whose territory the domain estate is, and is never fragmented into per-domain copies — affected domains pin it and render it by reference.

(SETTLED: first receipt 2026-08-28, hours after ratification — the fleet's decision register re-homed under its meta-concern through the six-step procedure, DEC-50 standing; the per-domain decision-sequence mechanics carry their own pre-registered receipt, the first domain-local ruling filed in its own spine. Ratified 2026-08-28 from the SaaS-registry derivation; standing receipts on the noun half — the fleet's shared-source register serving two readers so "what binds us" never forks, and one cross-domain ruling governing three concerns by pin.)

The re-homing rules

Stewardship moves, identity never does. RE-HOMING, when the true witness turns out to live elsewhere, is an induction-class act (on ruling, no clock) in two shapes, named by its first executor: CUSTODIAL re-homing moves stewardship AND address (the full six steps bind), while JURISDICTIONAL re-homing changes only the governing spine — slug, codes, and custody survive, so the alias and handover steps are satisfied vacuously rather than skipped, and the executor should know that lightening is legal.

In either shape: rows keep their codes, the vacated address becomes a PERMANENT one-hop alias (append-only records cite it forever, so its reader set is never provably empty and the retirement path can never legally reach it; vacated slugs are never reused), and the incoming owner verifies the inherited set against their own witness stream before accepting, backfilled from ground truth, never reconstructed.

Deletability is decided by PROVABILITY OF ZERO READERS: a same-cycle mint error nothing ever cited may be cancelled and its slug freed; anything that ever drew breath in the record may not. Re-

homing frequency is itself a health metric: a handful a year is an ontology learning, a pile is an ontology thrashing, and then the grain is the disease to treat.

(SETTLED: DEC-50, the DEC-register re-homing run through the six steps hours after ratification — the code sequence continuing unbroken THROUGH the re-homing as its own identity proof; the two shapes named by the first executor's friction note.)

Part 1.25: The two loops (where enforcement and gardening actually live)

Enforcement and gardening are not layers, and "on top" is the wrong preposition (this document's earlier phrasing). A layer is structure; these are FEEDBACK LOOPS, closing at different speeds over the three layers:

- **The fast loop (enforcement)** closes the runtime against the design: do the records conform to what the schema and SOPs intend? Its machine half is authored INTO layers 1-2 (state machines, required fields, gates); its human half is the Inspect ceremony walking the exception set. Cycle time: whatever clock the structure declared (a daily standup, a Monday walk).
- **The slow loop (gardening)** closes the design against reality: do the schema and SOPs still fit the drifted world? Cycle time: amendments, weekly to quarterly. It is the only loop allowed to reopen layers 1-2.

This is Argyris's single/double-loop learning in operational clothes (Part 5.5), and it dissolves a confusion the layer picture creates: asking "which layer does the walk belong to?" is asking which STRUCTURE a LOOP'S CARRIER lives in. Loops are behavior over the layers, carried by ceremonies like any other process (Part 2.5).

The reification principle

In a platform that implements this model, the definitions themselves are records. Layers 1 and 2 live as rows in meta-registers (the register of registers, the register of ceremonies, the charters), so the slow loop runs as ordinary register operations: an amendment is a row, a retirement is a kill-docket entry, and the gardening ceremony walks the meta-registers exactly as the standup walks the domain ones.

One discipline rides with the promotion: the meta-register is the LAST meta level; nobody ever proposes the register that manages the register that manages registers. Which platform implements this is a deployment decision, recorded where deployments are tracked, never in this canon.

(SETTLED: first receipt 2026-08-28, exactly on its pre-registered condition — the fleet's meta-concern, a register of registers entered through the pain door after one night of ad-hoc estate-gardening supplied its jobs with hours-old receipts, went live, ceremony-walked, with a grain-reversal amendment chain having run through the declaration machinery as ordinary operations. Ratified 2026-08-26; de-productized per review v4; promoted 2026-08-28.)

Spec and implementation

The spec owns intent, the implementation owns mechanism, and implementations conform to the spec. A structure's definition and its mechanisms rarely live in one artifact. The SPEC organ (a dossier, a charter, this canon) owns INTENT: names, legal states, contracts, clocks-as-commitments, decisions, and why. The IMPLEMENTATION organs (code, tools, configured platforms) own MECHANISM: the exact validation, retry logic, plumbing, which the spec never restates (prose restating mechanism is the mirror that drifts).

Declare the conformance direction at ontologize time: implementations conform to the spec, so a divergence is either an implementation bug or evidence the spec needs amending, and a human ruling decides which — the fast loop fixes the first, the slow loop the second. Implementation-as-SSOT collapses both loops into "whatever runs is right": a bug becomes indistinguishable from a decision.

Where one definition must exist in both prose and machine form (an enum, a type table), compile both consumers from one machine-readable source (config-as-data) instead of maintaining the pair by hand. Prior art: desired-state reconciliation (Kubernetes, Terraform: declared spec, controllers reconcile actual toward desired, humans amend the spec) and normative-language specs (RFCs).

(SETTLED: first receipt 2026-08-27, the email-intake concern's court read-state ruling — the pass implementation marked flagged court items read where the spec's enumeration implied they stay unread; the human ruling resolved it as an AMENDMENT, read-state redefined as "triaged by the pass" with the court pushing through the daily report line rather than the unread badge, and the spec now states the choice explicitly with a pre-registered flip condition. The law ran exactly as written: divergence found by conformance sweep, bug-or-amendment put to a human, spec amended, drift now impossible to repeat silently. Ratified 2026-08-26, from the email-intake design session's spec-versus-implementation question.)

Part 1.5: What success looks like

Before, a day in the life of an unstructured function: the lead's Slack is the intake, so they are interrupted hourly; every instruction is a meeting because there is no vocabulary to compress it into; the standup (if it fires at all) is a status recital with no sections, so decisions leave without owners; experiments run, learnings live in someone's head, and the next quarter re-runs them; when the lead travels, the function idles; and no agent can help, because there is nothing shaped for an agent to hold.

After: **the daily instruction is one line** ("EXP-044: run the lookalike variant, verdict Friday").

Ceremonies fire on clocks and run without the lead present (the absence test passes: two weeks of holiday, nothing rots). **The lead's attention is spent only on exceptions**, because compliance is silent. **Learnings accrue in registers** that new hires read on day one. **Agents carry the floor**: intake is gated, the standup is pre-briefed, the register lints itself. And the lead's calendar shows it: hours moved from chasing status to making the few calls only a human can make.

Measurable markers, baselined at the workshop and read at the practicum (day 30 formative, day 60 certified): average instruction length (should shrink to a line), ceremony survival rate (fired on cadence without prompting), exception-to-noise ratio in what reaches the lead, register growth (learnings written per week), and the absence test — which the practicum's unwatched weeks 5-8 administer for real.

Part 1.75: Authors and operators, and when structure pays

A well-run company is not a company of intuitive ontologists. Ritz-Carlton's average manager never authored anything: Horst Schulze authored the service ontology once (the Gold Standards, the \$2,000 pre-authorized empowerment rule: a defined enforcement boundary), and everything since is trained OPERATORS running an inherited stack, with the daily lineup as the ceremony and QA audits as enforcement. McDonald's is the extreme: a franchise manual so completely externalized that teenagers operate the restaurant. Mature industries commoditize the authoring: telcos import eTOM, banks inherit settled banking ontology plus regulation (enforced ontology imposed from outside). A company in a NEW category has nothing to import; its leads must author, and **authoring is the rarer**

skill this module trains: most corporate leadership programs only mint operators. The caution: inherited stacks rot without gardeners; big-company bureaucracy is an ungardened ontology, and the genuinely great operators institutionalized the gardening itself (kaizen is Toyota's gardening as a standing institution, and it is the half every Toyota cargo-cult skips).

The stage curve

When structure pays: the variable is coordination surface (people x functions x handoffs), not the funding round; rounds just correlate with it:

- **Founder-scale (Seed): hustle is correct, not primitive.** Everyone is in one room; the founder IS the ontology and walking enforcement. Heavy structure here is negative EV by this document's own fitness tests: the domain pivots weekly, the ontology would be gardened to death, the docket is empty.
- **Operator-scale (Series A): one tacit operator suffices,** applying structure just-in-time from intuition. Fragile: bus factor of one, and a trap: the operator's excellence DELAYS externalization, because everything works while routed through them, so nobody feels the missing structure until the operator saturates, and then it arrives as a crisis.
- **System-scale (Series B+): only externalized systems scale.** Either every executive carries the skill, or the stack is institutionalized (authored structures + trained operators + enforcement + gardening).

Two amendments: **agents shift the curve left** (enforcement no longer costs executive headcount, so a Series A company can afford Series B institutionalization early), and **a moat buys tolerance for mess** (structure is key to compounding, not to survival: a company with brutal switching costs can export its operational mess to its customers for years, paying in growth rate and customer resentment rather than death, and that resentment is the leading indicator of how moated incumbents eventually get displaced). Survivorship applies: the messy survivors are visible, the messy dead are not.

(SETTLED: founder-scale hustle correct at seed; agents-shift-left running in both estates.)

Construction grounds

Working ontologies are always EXTRACTED from evidence; failing ones are always speculated.

The top-down-versus-emergent ontology debate dissolves on inspection: the real choice is which evidence ground fits the domain. A SETTLED, EXTERNALIZED domain (a regulation, a reference model, twenty years of source systems) is imported and pruned: top-down is honest there because it is archaeology, not invention. A NOVEL, INTERNAL domain (your own operations) is extracted door by door, because its true structure lives distributed and tacit in the operators and is revealed only through use. The committee ontology grounded in neither is the form that fails from every direction.

Enforcement scales the stakes: the stronger the gates, the more receipts-first the authoring must be, because a wrong ontology under weak enforcement is ignored, and under strong enforcement is inflicted.

Top-down keeps exactly one legitimate role in an extraction shop: THE COVERAGE WALK, a periodic audit ceremony that enumerates the territory, marks which domains are committed, and consciously accepts the unmanaged remainder. The map is walked top-down; nouns are never minted there. An accepted gap is a decision; a silent one is a risk.

(PROVISIONAL: distilled 2026-08-27 from the Palantir-versus-doors first-principles sitting; promotion on the first import-ground domain shipped.)

The authority law

Authoring authority is GRANTED, RECORDED, and SCOPED, never assumed. Operators need no license: writing rows, running ceremonies, and walking exceptions stay open to everyone. Authoring — defining layer-1 nouns and layer-2 verbs, which enforcement then inflicts on everyone downstream — requires a recorded grant, and the grant has TWO paths: the APPRENTICESHIP — a reviewed first authoring, where the unlicensed author walks the door with a licensed one same-day and the pass mints permanent self-serve, so the license dissolves itself author by author at the pace real authors arrive (the fleet's working path, and the lighter of the two) — and CERTIFICATION, the practicum's day-60 pass minting an Author. Either way the grant lands in an authority register (who may author, in which domain's scope, granted when, on what receipt).

Two guards keep the license from becoming a priesthood: the pain door is never blocked (the unlicensed propose through agents, which enforce the admission standard mechanically; a licensed Author ratifies the apply), and the licensed path must remain the easiest path, or the org routes around it into shadow structure.

(SETTLED: the first authority register ran at a live write path 2026-08-28 — the fleet's concern-apply allowlist, ten frictionless hours on day one, with the two reviewed births producing the estate's two best declarations.)

Part 2: The three skills (what actually separates good from bad)

The layers are anatomy. The skill is physiology:

1. **Diagnosis: seeing an incident as a missing noun.** Chaos is never "people not trying hard enough"; it is a structure that does not exist yet. Interruptions plaguing the team = there is no Intake. Learnings evaporating = there is no Experiment register. The unskilled manager works harder inside the chaos; the skilled one asks "what noun is missing?" Think in classes, not instances.
2. **Design taste: which ontology, not just an ontology.** Two managers both build a standup; one's sections mirror real decision points, the other builds status theater. The fitness tests in Part 4 are the review side of the difference; the design side has a procedure: **enumerate the jobs before naming the nouns**. List who will hire the structure and the questions they will bring to it ("what is late?", "who has the ball?", "have we already learned this?"), then derive the nouns and fields from the jobs. Ontology engineering calls these competency questions (Part 5.5); product calls it JTBD. It converts "every field changes a decision" from a filter into a generative method: fields are born from decisions instead of deleted for lacking one.

Guard, the admission standard: a job enters the enumeration only by stating four things — WHO hires it (a person or process by name, never "we"), on WHAT CADENCE it fires, WHICH DECISION its answer changes (else it is curiosity, not a job), and its FAILURE COST when unserved; speculative jobs are how the 22-field form gets built. The direction of derivation is the method's spine: jobs then structures, never structures then jobs — writing a job to bless a structure you already want is post-hoc rationalization, the disease pre-registration exists to kill. And the door stays open after birth: a job discovered late enters by amendment with full honors — gardening is the JTBD door's safety net, a late job is not a failure, an UNNAMED one is, and the enumeration is never finished at birth; it is finished never.

This gives Ontologize two doors: the PAIN door (the catechism, Part 3: a live domain, entered through recurring pain) and the JTBD door (a greenfield domain, entered through job enumeration). Two sizing rules ride the doors (ratified 2026-08-27, from the first live teach): THE DEPTH RULE — the doors do not just start the loop, they bound it: ontologize only as far as needed to solve the pain or satisfy the jobs, never for ontology's sake ("should I ontologize strategy?" has a one-question

answer: which door?); and THE SPLIT RULE — a committed domain whose enumeration passes roughly ten jobs is two domains wearing one name; split it.

(Renamed from incident/survey at v3.0.0: the field-tested names from the first live teach, 2026-08-26, displaced the bookish pair before any cohort learned them. Admission standard, retro-fit trap, and late-entry clause ratified 2026-08-26, sharpened on the J8 case, where a summary-corpus proposal failed the guard and its real job then entered by amendment with a serving path.)

3. **The two loops: enforcement and gardening.** Structures die of non-use, and bad design is the leading cause of non-use (a 22-field form does not get "enforced into" adoption; it gets routed around). A structure is alive only if skipping it is loud (enforcement) and someone amends it when reality pushes back (gardening).

The grain law

A committed domain is a managed PIPELINE, not a category and not an outcome. The split rule caps size by count; this law tests it by KIND, on a ladder. A CATEGORY ("Marketing") is too big: no single owner, no closable noun-set; its spine would be an org chart, and nothing can conform to an org chart (that axis belongs to responsibility, not territory). An OUTCOME ("Retention") is not a domain either: it is a KPI axis; you do not walk an outcome, you walk the programs that move it (outcomes belong to goals and scorecards). A committed domain is the PIPELINE underneath: one owner, a closed vocabulary, real records moving.

Four portable tests: (1) ONE accountable owner — two owners means too big; (2) a closed noun-and-verb vocabulary that fits one spine page — if you cannot close it, too big; (3) two or more admitted jobs with DIFFERENT named hirers — one job serving one hirer through one register is an ORGAN, and an organ lives inside some domain, never as one; (4) the spine test — a divergence between the spine and a running organ must yield a question a human can rule on ("Marketing diverged from reality" is unrulable; "a redemption row missing from the reconciliation" is a Tuesday).

And the capacity guard that binds both directions: every committed domain costs a walk, so the domain count is bounded by real inspection capacity — lumping starves clarity, slicing starves the walkers.

(SETTLED: distilled 2026-08-27 from the fleet's first sizing ruling (DEC-29, the grain ladder) — a gift-card pipeline admitted, "Marketing" and "Retention" refused — and promoted 2026-08-28 on its pre-registered receipt: the second independent sizing ruling landed on the ladder unamended, its author running the exclusion analysis herself and stamping the one genuinely uncertain boundary job provisional with its own receipt-due condition, the grain machinery absorbing what a premature call would have forced.)

Part 2.5: The five verbs (the practice)

The layers, projected into what an author-operator actually DOES to a domain. This is the sequence students should carry in their heads:

#	Verb	What it does	What it projects
1	Ontologize	Name the nouns: entities, fields, legal states, declared places	Layer 1 (authoring)
2	Processize	Define the verbs: SOPs, ceremonies, cadences, owners (including the two meta-ceremonies below)	Layer 2 (authoring)
3	Log	Every event lands as a row, as it happens	Layer 3 (runtime)
4	Inspect	Walk the exception set the machine surfaces; disposition every item	The fast loop's human half (runtime)
5	Garden	Amend the ontology and SOPs as reality pushes back	The slow loop (runtime)

Two teaching points, each load-bearing:

The list is two kinds of things, not one pipeline. Verbs 1-2 are authoring acts: one-time transformations of the domain, revisited only through Garden. Verbs 3-5 are runtime FUNCTIONS the running system must perform, each carried by processes authored in step 2: Log is carried by the "write the row" clauses inside every SOP, Inspect by the walk ceremonies, Garden by the gardening cadence. The classic objection ("the walk's SOP is defined in Processize, so isn't Inspect just part of step 2?") applies equally to Log, and that is the tell: 3-5 name duties whose PRESENCE the deployment is audited on, not additional things to author. The suffix asymmetry is deliberate: you -ize twice, then you run. ("Processize" is a deliberate coinage mirroring Ontologize; the dictionary neighbors are proceduralize and operationalize, and the symmetry was ruled worth the coinage, 2026-08-25.)

Inspect and Garden are meta-processes. Log operates on the domain (deals, sessions, campaigns); Inspect and Garden operate on the system itself: Inspect carries the fast loop's human half (runtime state versus intent), Garden carries the slow loop (schema and SOPs versus drifted reality; the loop model is Part 1.25), and Garden is the only runtime verb allowed to reopen layers 1-2. The system manages the work; two of its processes manage the system. A deployment that processized only domain verbs, and forgot to processize its own inspection and gardening ceremonies, is exactly the deployment that rots.

(Ratified 2026-08-25. Born from the Inspect-versus-walk question; the observation that the walk's SOP lives in Processize is what forced the authoring-acts versus runtime-functions distinction.)

Two kinds of processes

A process is either a domain-owned SOP or a ceremony, and forcing a ceremony into one domain mints a junk drawer. A DOMAIN-OWNED SOP belongs to exactly one managed domain: authored in that domain's SPINE (the term of art for a domain's governing spec document, whatever its local name: a dossier, a charter, a concern page), one owner, the ownership invariant strict. A CEREMONY is a cross-domain CONVENOR: it walks segments that several domains each own (a standup, a weekly walk), so it is owned at the level that spans what it convenes and is never forced into a single domain (forcing it produces the junk-drawer domain). Ceremonies are how Inspect runs ACROSS domains: one sitting reads each domain's exception segment.

The diagnostic is itself useful: a process nobody can assign to one domain without inventing a fake one is a ceremony, and that is information about the ontology, not a defect in it.

(PROVISIONAL: first receipt logged 2026-08-26, the standup-prebrief diagnosis; promotion on the next independent case. Ratified 2026-08-26 from the estate's concern-model sitting, where a ceremony's "missing owner" turned out to be the model missing this category.)

The spine anatomy

A committed domain's spine composes, whatever its local format: the SCOPE sentence (what this domain is and when to open it), the REIFIED JOBS (the admission-standard table), the ONTOLOGY declarations (nouns, legal states, identity schemes), the SOP SUITE (each verb named, clocked, owned), the ORGAN INDEX (typed pointers to the registers, tools, and workflows that implement — mechanism never restated), the PINNED DECISIONS (rendered by reference, per the jurisdiction law), and a VERSION.

Corollary, the orphan rule: an organ without a spine home is an orphan. At build time the question "which domain does this organ serve?" is MANDATORY, with three legal answers — an existing domain covers it (build under it, back-link); the domain is real but undeclared (declare it through a door as the FRONT HALF of the build, jobs-first); or the build is WORKSHOP, explicitly experimental and uncataloged, earning structure by receipt. Only the silent fourth path is forbidden: a standing, cataloged organ with no answer at all.

Enforcement climbs the ladder on evidence — ask at birth, notice after, gate only when coverage makes gating honest rather than performative, because a day-one no-domain-no-build gate inverts the method into permission-first bureaucracy and mints junk domains written backwards to bless wanted builds.

(SETTLED at birth: two implementations converged on it independently — the estate's thirteen dossiers and the fleet's eight declared concerns, across four different authors. Ratified 2026-08-28. Provenance, per the double loop: this REVERSES the 2026-08-26 ruling that the composed pattern should live as examples only; that ruling's blocker was the then-ungated authority dimension, which has since earned its receipts and its law.)

The induction law

Between authoring a domain and running it lies the INDUCTION: the one-time act carrying the domain's unstructured past into the standing process's steady state. The two process kinds do not name it. It has an owner and a declared end state ("caught up", defined in advance) but NO clock: it runs once.

Three legal treatments of the past, mixable per domain: BACKFILL (enrich from retained ground truth — statements, archives — so the past becomes rows under the new ontology), FREEZE (a dated legacy line: the past stays readable but unconverted, and the new capability stays quiet behind the line, which is correct, not a gap), and BANKRUPTCY (the past declared handled as a class without item-level processing — legal only when reversible by construction: archived, never deleted, the corpus retained). The governing constraint on all three: records never mutate, derived views may recompute everywhere, and the past is enriched only from real ground truth, never reconstructed.

A second induction on the same domain is not an induction; it is evidence the standing process's enforcement failed and backlog re-accumulated: diagnose the fast loop, then sweep.

(PROVISIONAL: three receipts on record — an email bankruptcy sweep, a brokerage backfill, a day-ledger's frozen legacy days; promotion when the next induction fits without amending the taxonomy. Ratified 2026-08-26; the vocabulary is the author's own "onboarding to the process", born from the EMLSOP-1 question.)

Inspect means exceptions

Inspect means exceptions only (the Deming note). Deming's point 3 condemned depending on END-OF-LINE mass inspection, and his remedy (improve the process itself) is this model's Garden; Inspect survives his critique precisely because enforcement (machine-run invariants, authored in steps 1-2) builds the quality in upstream, so what reaches the human is only the exception set: overdue, blocked, stale ball, no next action, failed assertion.

Walked on cadence, as a tick-down, every item ending in exactly one recorded disposition: do, delegate, defer, drop, or decide. A manager inspecting everything is the enforcement layer failing (the two-hour, 150-item standup is this failure mode with a calendar slot). And a recurring exception is information about the system, not about the work: route it to Garden instead of re-handling it heroically each week.

| (SETTLED: the 150-item-standup failure mode named and avoided; the email court's disposition set completed by defer.)

The five presences

A corridor screen, never a gate. The catechism (Part 3) authors a structure; this is the quick screen for a live one, run in a hallway minute. The GATE instruments remain Part 6's (the scorecard at design time, the practicum's receipts checklist at day 30/60; where they overlap, the gate wins).

For any register, five questions: (1) Ontologized? Show the schema, and does any field still change a decision? (2) Processized? Show the SOPs, including the meta-ceremonies. (3) Logging? Show rows landing. (4) Inspected? Name the ceremony that walks its exceptions, and its cadence. (5) Gardened? Name the last amendment and its evidence, or the honest null (nothing creaked, and the monitoring that would catch it is armed).

Five yeses is a screen passed, not a certificate: a backfilled register can fake presences, and only the practicum's decision-with-artifact test proves life.

Part 2.75: The organ types (one language across deployments)

A domain's spine points at its organs (the organ index, Part 2.5), and until now every deployment named those organs in its own dialect: the estate says dossier-table, the fleet says tracker, and a conversation between the two needs translation. This Part fixes the vocabulary: a closed set of organ TYPES, each named by the one JOB it does for a domain. The set was EXTRACTED, never designed (the construction-grounds law): a coverage walk over the two living deployments' actual inventories — the personal estate's substrate and know-how dials, its registers, mini-apps, and ledgers; the fleet's concern declarations, tracker engine, boards, and receipts lanes — kept the common types and marked the divergences.

Two boundaries before the table. The SPINE is not an organ: it is the organ-HOLDER (the spine anatomy, Part 2.5), and organs are what its organ index points at. A CEREMONY is not an organ either: it is a layer-2 process (Part 2.5); its derived docket binds as an index and its walk record as a ledger.

#	Type	The organ's one job	Estate binding today	Fleet binding today
1	The register	The domain's records: identified rows, mutable cells, exceptions walked	dossier-table, Notion database, Eloquent models	engine tracker, workshop-lane register
2	The ledger	Append-only receipts: proof that events ran, never mutated	the Alfred ledger, the email manifests	the access-change ledger, workflow run receipts
3	The court	Where balls sit, visible at a glance	inbox-as-court, with snooze and resurface	the Relay court
4	The board	A bounded body of work in flight: tasks, definition of done, closable	the nws-todo item tree	the Relay board
5	The surface	The screen a human opens; view-only is a qualifier, never a noun	the mini-app	the tracker surface, the tier-3 dashboard
6	The door	The typed write path: Gate-mode enforcement embodied	MCP tools, the email actuators	the concern-apply door, engine row writes
7	The index	Derived views recomputed on events, never kept by hand	the dossier index	the concern index, ceremony dockets, coverage views
8	The corpus	Retained raw ground truth for recompute and backfill	the email corpus, statement PDFs	transcripts, harvest corpora

The binding columns are ILLUSTRATION, not law: they record where each type ran on the day of extraction, and they will drift as deployments evolve, which is legal — that is the point of the law below. Two organ shapes stay deliberately deployment-local: the SCOREBOARD (the fleet's goal layer: WIGs, lead measures, readings; the estate runs none) and the FORM (the estate's printed and iPad fill-in sheets; agents print nothing). Each is a legal local noun with a standing flip condition: it enters this table the day a second deployment earns it by receipt.

Types and bindings

The canon owns organ TYPES; a deployment owns only BINDINGS. A type names the organ's job, exactly one per organ, and lives here (machine form: `canon/ORGAN-TYPES.yaml`, the config-as-data source deployments bind against — deliberately carrying types only, since a binding listed in the canon would violate this very law). A binding names the type's substrate in ONE deployment — the register binds to a dossier-table on the estate and to an engine tracker on the fleet — and lives in that deployment's own declarations, amended freely on its own clock.

New vocabulary enters a deployment as a binding of an existing type; a shape no type fits stays a deployment-local noun with a pre-registered flip condition, and is canonized as a new type only when a second deployment earns it by receipt (the scoreboard and the form are the standing examples). This is the spec-and-implementation law applied to vocabulary itself: the canon is the spec of the language, deployments are its implementations, and a conversation about any estate Norman runs needs no translation.

(PROVISIONAL: extracted 2026-08-29 from the two living deployments' inventories and ratified the same day on Norman's one-language directive, NW-ANA-22. Promotion when the next deployment-scale binding lands on the table unamended: Relay's tenant-facing model, or the next organ birth in either estate.)

Part 3: The catechism

Run this every time a pain recurs twice. Write the answers down; they ARE the design. (This is the PAIN door into Ontologize, for domains already running and hurting; a greenfield domain enters

through the JTBD door instead, the job enumeration in Part 2, skill 2.)

1. What is the recurring pain?
2. What noun or place is missing? (layer 1: define its fields, its legal states, its ID scheme, where it lives)
3. What is the one verb, on whose clock? (layer 2: a cadence with an owner; "sometime" is not a clock.
One DOMAIN verb: the meta-ceremonies ride your existing walks until the structure earns its own)
4. What does one row look like, who writes it and when, and what does empty look like? (layer 3: the row-writing moment is Log's home; empty must degrade gracefully; "nothing due, closed in 30 seconds" is a success)
5. What makes skipping it loud? (enforcement)
6. When it creaks, who amends it, and how? (gardening)

Part 4: The fitness tests (the review rubric)

A proposed structure passes review only if:

- **Every field changes a decision.** If no decision differs based on a field's value, the field is decoration. Delete it.
- **Every instance has a declared identity.** A minted, permanent, speakable ID where instances are cited across time, people, or surfaces; a declared natural key where the date or slot individuates (the identity law, Part 1). A structure whose instances can only be referenced by description ("the campaign Budi ran in March") fails: claims cannot carry receipts and dispositions have nothing to attach to.
- **Compliance is silent, exceptions are loud.** The structure surfaces only what needs a human; green rows make no noise.
- **Pull surfaces die.** Anything that waits to be opened will not be opened. Every structure needs a push moment: a clock, a ceremony, an alert.
- **A good ceremony shrinks.** Anything appearing identically every cycle is an automation candidate. A structure whose docket trends empty for a quarter merges into something larger. A structure that can never end is a treadmill.
- **The instruction test:** after the structure exists, can the manager direct a day's work in one line? If not, layer 1 is too coarse.
- **The absence test:** does the structure run correctly for two weeks with the manager on holiday? If not, layer 2 has no real clock or no real owner.

The cognitive-fit law

Structures are sized to the OPERATING HEAD, never to the domain's ambition. Working memory holds roughly four to seven chunks, and recall descends labeled hierarchies (organized lists recall at multiples of flat ones — Bower), so every surface a human must hold AT ONCE is capped at a human span: the instruction fits one line, the vocabulary fits one spine page, the unsure list fits five, the routing sentence fits forty words, a hierarchy node fits about seven branches, a domain's jobs fit about ten (the split rule is this law at the jobs tier), the domain count fits the walkers (the walk-capacity guard), and the whole discipline fits a ten-word mantra.

When reality outgrows a surface, SPLIT or DESCEND A LEVEL — never stretch the surface. This is the one law beneath half the tests above, and the canon's own register obeys it (seven models, seven instruments, seven families of at most five laws).

Prior art: Miller's and Cowan's spans, Bower's hierarchical recall, Krug — whose don't-make-me-think is this law's surface-side twin in the design canon — and org theory's century of independent

convergence: span of control at 5-9, squads at 8-12, two-pizza teams. The convergence is the tell: every discipline requiring one human to hold a set of accountable things lands on the same band, because the constraint was never the domain. It was always the head.

(SETTLED at naming: each of its instances already carries receipts; named 2026-08-29, when the ~10-job split line was reverse-derived from working memory.)

The qualifier corollary

A variant is a QUALIFIER on an existing noun, never a new noun, until receipts earn the split (the cognitive-fit law applied to vocabulary growth). A screen-shaped organ that owns no tables is a "view-only mini-app", not a "view": the vocabulary stays one noun wide, the qualifier carries the variance, and a first-class noun is minted only when accumulated qualified instances demonstrate jobs of their own (the split rule, applied to the dictionary instead of the domain). The organ type table (Part 2.75) grows the same way: bindings and qualifiers first, new types only by second-deployment receipt.

(SETTLED: first receipt the view-only mini-app ruling, 2026-08-29, recorded the same sitting in the estate charter, the doctrine repo, and the claims dossier; the type table's own admission rule is this corollary in standing practice.)

Deferrals carry flip conditions

Kill conditions govern live structures; their constructive mirror governs structures NOT built.

When review rejects or defers a proposal whose job is real but already served more cheaply, the rejection is pre-registered as a FLIP CONDITION: the observable event that reopens the build ("the day a real query fails the live serving path"), with the build then designed for the failing case specifically and sized by its evidence.

Structure is earned by receipt, never by scenario; a deferral without a flip condition is a door slammed, and the flip condition is what makes saying no cheap.

(PROVISIONAL: receipts, the J8 search-corpus deferral and the estate's borrowed-substrate rulings; promotion on the first deferral that returns through its condition.)

Part 5: Agents as enforcement

In code, enforcement is free: the compiler and the database sit in the write path and make illegal states unrepresentable. Organizations never had this; the only checkers were humans, too expensive to post at every write path. **Agents change the economics.** Four strengths, in order (vocabulary unified 2026-08-22 across the canon, practicum, and product thesis; the fleet charter independently named "the noticing layer" as its own organ, and same-day alerting is not linting):

- **Observe (CI):** an agent attends the ceremony or reads the channel and lints outputs against the ontology after the fact: did the standup cover exceptions? did the decision leave with an owner and a date? Violations still happen but become loud, eventually.
- **Notice (the same-day alarm):** an agent watches the clock and the write path live: a ceremony that did not fire, a row gone illegal, a due date crossed — surfaced the same day, to the owner, once, loudly. Detection at the speed that prevents the 29-day silent death.
- **Gate (compiler):** the agent IS the write path. Work enters through a typed door that stamps the required fields and cannot produce an illegal row. Nothing can vanish silently because the only way in is through the door.
- **Scribe:** the agent produces the artifact natively in the right shape (pre-briefs the standup, files the register row from the conversation). Compliance becomes moot. (Renamed from "Author" 2026-08-

26: that word belongs to the authoring acts and the certified Author. Under the five-verb taxonomy this mode is the Log function performed by an agent; it stays in this list because making compliance moot is the strongest enforcement there is.)

The accountability primitive: the ball and the court (nouns adopted 2026-08-22 from Relay, where they run in production with a ledger). "Decisions leave with owners and dates" needs a noun to be enforceable: the **ball** is who owes WHAT, since WHEN, held by WHOM, verified against ground truth before anyone chases; the **court** is where every ball currently sits, visible at a glance. A decision without a ball is a wish with a witness; the noticing layer's core job is balls going stale.

Three cautions, all load-bearing:

1. **Enforce artifacts and write paths, never conversations.** Police how people talk and they will route around you into shadow channels. Enforcement holds when the enforced path is also the easiest path: the typed door must be more convenient than the workaround.
2. **Enforcement strength must be matched by gardening capacity, and calcification risk must be located precisely.** Calcification does not live in the mere existence of automation; it lives in three places: IRREVERSIBLE ACTIONS (a machine verb that touched the world cannot be re-run differently), NON-RECOMPUTABLE DERIVATIONS (stamps whose inputs were not retained), and CONSUMER COUPLING (rules, habits, and downstream structures written against the current vocabulary). Where the corpus is retained and derivations are recomputable, iterating the ontology stays cheap after mechanization: version it and re-derive wholesale (derived views may improve everywhere; records never mutate).

The corollary: **mechanize observation before action.** Stamp and classify at full volume immediately, because it accelerates the slow loop (the whole distribution becomes visible instead of a hand-walked sample), but each acting verb graduates on its own ladder: suggest, then act-reversibly, then act-irreversibly, each step earned on measured precision. One irreversible mistake during ontology churn burns the operator's trust, and distrust kills the system faster than calcification ever would.

One scoping caveat that binds hardest in this course: the observation corollary was derived from a corpus of MESSAGES; when the observed are PEOPLE (this course's runtime), observation graduates on its own ladder too — artifacts before behavior, aggregate before individual, disclosed before silent — or the full-volume stamping IS the surveillance that caution 1, the performance firewall, and the power question all warn about. (Refined 2026-08-26, the email-intake case; human-runtime caveat per review v4, finding 3.)

3. **Derivation before delegation.** Never seat a model-driven keeper over what pure derivation computes: an index, graph, or scoreboard derivable from the declarations should be recomputed on write events — never stale, nothing to babysit, nothing hallucinated — where an agent doing the same job can lag, die, or invent. Agent personhood is earned only by the JUDGMENT layer (semantic collisions exact-match cannot see, drift noticing, gardening proposals), and a judgment agent PROPOSES to the owning authors, never mutates a declaration: an agent that rewrites ontologies is an unlicensed author by the authority law. (Ratified 2026-08-28, from the fleet dissolving its proposed ontology-keeper service account into console-side derivation plus a proposals-only noticer.)

Honest scope: agents enforce **shape** essentially perfectly (fields present, receipt attached, disposition recorded). **Substance** (was the decision good) stays human review. Do not oversell.

Part 5.5: Prior art (and what is actually new)

This model has a 40-year pedigree, which is a feature: it is engineering with a literature, not a founder's invention.

- **The layers:** Zachman Framework (1987, explicitly an "enterprise ontology"), TOGAF, and the BPM guild (BPMN, the BPM CBOK). Whole industries have shared, consortium-authored ontologies: eTOM for telcos (TM Forum), SCOR for supply chains, FIBO for finance, HL7 for healthcare. A mature industry is one where layers 1-2 are commoditized and importable by hiring; a company in a new category has nothing to import and must author.
- **The operating loop:** ISO 9001 (process + external audit = enforcement), Toyota (kaizen = institutionalized gardening), Grove's *High Output Management*, Bossidy & Charan's *Execution*. The closest single ancestor of this document's full stack is Amazon's **mechanisms** culture: "a tool plus adoption plus inspection plus iteration" maps one-to-one onto ontology+SOP, enforcement, and gardening.
- **The intellectual umbrella:** Herbert Simon's sciences of the artificial (design science), Stafford Beer's management cybernetics, Nonaka's externalization of tacit knowledge — and Argyris's single/double-loop learning, which is exactly the enforcement/gardening pair: enforcement is the fast loop (per event: does reality conform to the structure?), gardening the slow loop (per epoch: does the structure conform to reality?). The loop model (Part 1.25) has siblings in control theory's inner/outer loops and IBM's MAPE-K autonomic manager (monitor-analyze-plan-execute over a knowledge base), the closest software ancestor of agents running both loops over reified definitions (its Plan/Execute is adaptation, the slow loop's work, not only checking).
- **The design methods:** Grüninger & Fox and Uschold's competency questions (1990s ontology engineering: an ontology is adequate ONLY IF it answers the questions enumerated for it — a requirements test, not a sufficiency proof), which the JTBD door (Part 2, skill 2) rhymes with from the product side in Christensen and Ulwick's jobs-to-be-done (they theorize purchase motivation, not schema adequacy; the rhyme is real, identity is not claimed). Behind the identity law sits the spirit of Quine's "no entity without identity" (his demand is identity CRITERIA — when is x the same entity as y — which minted codes implement but do not exhaust) and a practice lineage as old as coordination itself: folio and voucher numbers in double-entry bookkeeping, part numbers in interchangeable-parts manufacturing, patient MRNs, ticket and RFC numbers.
- **The platform-scale ancestor:** Palantir Foundry's Ontology (typed object and link types plus a KINETIC layer of governed action types as the only write path, with AI agents grounded on the ontology so an untyped write is impossible by construction) independently validates the layer model and Gate-mode enforcement at enterprise scale. The honest split: Foundry's construction is archaeology of settled source systems, correct for its customer per the construction-grounds law (Part 1.75); its ontology MIRRORS operations that happen elsewhere, where this model's ontology RUNS them; and it carries no coordination primitives (no ball, no ceremony, no walked exception), which is precisely the layer this discipline adds.

Where this sits (the discipline coordinates)

Locate any discipline by three coordinates: object, method, unit of intervention. Ours: the **object** is the coordination substrate (the shared symbolic layer an organization steers itself by: registers, states, ceremonies, artifacts, and their enforcement — register-first, not process-first, which is why the nearest literature cousin is the artifact-centric process school, business artifacts with lifecycles, and the deepest cousin is Ostrom's institutional design principles: monitoring, graduated sanctions, cheap conflict resolution, nested amendment rights). The **method** is design science in Simon's sense, with receipts as the epistemology: build the artifact, certify it by unwatched operation, state falsification conditions, iterate. The **unit of intervention** is the line-manager-authored micro-structure (one register, one ceremony), against Enterprise Engineering's and TOGAF's top-down

architect-borne interventions on the same object. As a thought experiment, the object-level name would be **coordination engineering**; this course does not claim the name, for the reason below.

The honest self-test: a discipline needs a distinct object (arguably yes), distinct methods (partially: receipts + agentic enforcement), a community with shared vocabulary (no: one estate, one company), and cumulative results others build on (not yet). Two of four. Today this is a PRACTICE with a thesis; "discipline" is a title only outsiders confer (Toyota never named lean; MIT did, forty years later). The falsifiable milestones: the school exists the day someone never trained by the authors runs this vocabulary back at them; the discipline conversation begins the day an organization the authors do not own publishes receipts using it.

What is actually new here, in honest order of size:

1. **Agents as enforcement (the AI-era delta, and the big one).** Every prior framework's enforcement required expensive human audit bureaucracy, which is why only mature giants could afford compiled-grade structure, and why the frameworks so often decayed into shelfware: companies bought the ontology, could not afford the enforcement, and skipped the gardening. Beer tried to build cheap cybernetic enforcement in 1971 (Project Cybersyn, on telex machines) and failed on the technology. Agents make the checker nearly free at every write path; a Series A company can now afford what previously only a Ritz-Carlton or Toyota could institutionalize.
2. **Compression for teaching.** eTOM is hundreds of pages; TOGAF needs a certification course; nobody hands those to a marketing lead. The 3-layer kernel + catechism + rubric is a 45-minute teachable distillation.
3. **Exam-by-receipts.** The guilds certify recall; this program certifies a structure surviving 30 live days.

Part 6: The training program

Four elements. Two are assessment gates, precisely named: the exam checks you UNDERSTOOD; the practicum checks it WORKS (Norman's own ontology correction, 2026-08-20: a receipts review is a practicum, not an exam).

Element	Format	Content
Lecture	45 min, once	Parts 0-5 and 7 of this document plus the Part 8 close, and one real worked example shown failing and then fixed
Workshop	90 min, same day	Each lead brings their OWN recurring pain and authors the structure for it using the catechism, reviewed live against the rubric (drill #1). No toy cases: the domain knowledge is the point, and authoring is the skill being trained
Diagnostic exam (Gate 1)	~20 min, one synchronized sitting, end of workshop day	Tests the trainable skill, not recall: unscored warmups + fresh-bank diagnosis vignettes + the cold transfer probe. Pass rule, re-sit rule, and item pools live in ONE home: exam/DIAGNOSTIC-EXAM.md (restating gate parameters here is the E3 defect the course warns about — v3, C2). A fail is remediated same-day and re-sat once; a second fail defers the practicum to the next cohort
Pulses	Day 7 / 14 / 21, ~15 min each	The practicum's own enforcement layer (added after the v1 adversarial review found the 30-day silent run was enforcement-by-hope): agent-collected fired-counts where wired, async check where not, same-day nudge on any missed cycle, plus one vignette drill per pulse for spaced practice. Declared in the scheduling contract before anyone leaves the workshop, so nothing at day 30 is a surprise
Practicum (Gate 2)	Formative at day 30, certified at day 60	Days 1-30 are supported (pulses); day 30 reviews receipts formatively. Weeks 5-8 run completely UNWATCHED, and certification requires the structure to have survived them cold (v2 review: a structure kept alive by pulse pressure and a load-bearing one are indistinguishable under monitoring — so the course administers its own absence test to what it certifies). Pass = alive unwatched + gardened on evidence (amendment trail or honest null) + the instrumented team-side median not net-negative . Day-60 certification: a clean-room AI grader runs the committed, participant-readable protocol (exam/CERTIFICATION-GRADING-PROMPT.md) — context isolation removes authorship bias — and a human WITNESS outside the participants' reporting lines attests prompt-verbatim / one-run / verdict-unedited and delivers (operator control is the part isolation cannot fix; the witness is the firewall — v3 C3, resolved). A fail is a diagnosis, one re-review maximum; then the structure retires honestly and the participant restarts with a new pain

Format (Norman's ruling after the v1 review): **parallel 1:1 tracks, ~6 leads at a time**. All reviews, remediations, and pulses are 1:1, never in front of the room (the v1 public-review design predicted theatrical compliance when the reviewer is the CEO). Cross-pollination moves to the sanitized case library (where an anonymized FAILED structure earns a Distinction credit, so the library is not survivorship-biased by construction). Two preconditions gate scheduling: the sponsor's written **performance firewall** (course outcomes never enter performance evaluation) and **named, numbered fleet-gate capacity** (cohort 1 default: two gates). Disagreement protocol, stated at the workshop: the participant owns every domain call; the reviewer may only cite the rubric. Honest dose accounting (v2, recounted at v3): this module installs the vocabulary, TWO supervised authoring reps (the worksheet and Q11's cold transfer probe), and one 60-day operating-and-gardening run, plus recognition drills at the pulses; taste arrives with the second and third structures, which is what Distinction measures.

Note the self-hosting property, which is also the credibility argument: the lecture installs layer 1, the workshop runs layer 2, the practicum is layer 3, the diagnostic exam is the enforcement gate on the write path into the practicum, and the day-30 review is gardening. The curriculum is built from the model it teaches.

The module package (the definition of done)

The training is DONE when the module runs without its author, which means five artifacts exist:

1. **The reference doc** (this document): the model, skills, catechism, rubric.

2. **The lecture script** (45 min): Parts 0–5 and 7 plus the Part 8 close, with the worked failing-then-fixed example, openable by any trained reviewer, not only Norman. Metaphor kit for the script (chosen against the rule that a lecture metaphor's failure must be INTRINSIC to its domain, which killed the dance/choreography candidate: ensemble freestyle on a stage is merely an aesthetic choice): primary = **the kitchen brigade** (Escoffier's stations as nouns, the ticket as the row, service as the clock, the pass as a literal write-path gate, menu revisions as gardening; six talented cooks with no brigade on a busy night fail on coordination, not skill, and the food-truck-to-restaurant arc carries the stage curve); one-liner = **the rowing eight** (eight strong rowers without a stroke rhythm go in circles: strength is not the variable); evidence anchor = **the checklist studies**, WITH the replication caveat spoken aloud (the 2009 WHO trial's headline effect was strongest at lower-income sites and a large Ontario implementation study found no significant mortality change; the honest claim is "checklists help where implementation is real, and the population studied was elite" — saying this out loud costs 15 seconds and models the epistemic standard the course preaches); history slide = Part 8's lineage, presented explicitly as ILLUSTRATION, not evidence (survivorship cuts both ways: the Mongol succession collapsed within decades), with Rome/Genghis for "the system survived bad generals" and "speed FROM structure".
3. **The workshop kit**: the catechism as a fill-in worksheet, the rubric as a scorecard, and one page of facilitator notes (common failure patterns to catch live).
4. **The assessment set**: the diagnostic exam (Gate 1: fresh-item vignette bank + keys, agent-gradable, with re-sit spares held back) and the practicum protocol (Gate 2: day-30 formative receipts review + the day-60 cold certification check).
5. **The worked examples**: lifting + marathon (the domain-neutral pair) and one real Gotrade before/after once the first cohort produces it.

The module passing its own absence test (someone other than its author runs a cohort successfully) is the graduation of the training program itself from layer 3 to a standing institution.

The whole discipline, ten words, for the closing slide: **name it, clock it, write it, guard it, garden it.**

The mantra tracks the model, not the verb list: name/clock/write are the three layers, guard and garden are the two loops. "Guard it" deliberately commands the whole fast loop, both halves (author the gates, then walk only what escapes them); Inspect is the half that stays yours. A mantra that said "inspect it" would command the mass-inspection failure mode instead of the design that prevents it (the Deming note, Part 2.5).

Part 7: The honest scope

This is not a growth strategy; product, market, and distribution are (Part 0 makes the real claim: accumulation and agentization). Two more honest boundaries, stated up front because they build trust with a skeptical room:

- **Structure is not bureaucracy, and the fitness tests are the difference.** Every field must change a decision, ceremonies must shrink, and a docket empty for a quarter merges away. Anyone in the room can kill a field or a ceremony by showing it fails the rubric. Process that cannot be killed by evidence IS bureaucracy.
- **Agents enforce shape, humans judge substance.** The fleet can guarantee the experiment row exists with a hypothesis and a verdict; whether the hypothesis was worth testing stays a human call. This program makes judgment go further; it does not replace it.
- **The model is a vocabulary and a design discipline, not a predictive theory.** Any failure can be narrated post-hoc as "some layer died," which explains everything and therefore predicts nothing; own that. The falsifiable claims live at the INSTANCE level, and here is the standing one: a structure

that passes all seven fitness tests and still changes no decisions after 60 live days indicts the model's fit for that domain, not the participant. When that happens, say so and retire the structure.

- **Structure the pipeline, never the ideation.** Brief, review, launch, learning: structure them. The thinking that produces the campaign concept, the product idea, the design: that is exploratory work, and Norman's own doctrine says formal process kills it. If the noun you are about to define is an idea, stop.

And the restated aphorism, for the room that believes "execution eats strategy for breakfast": three things get conflated in that phrase: strategy (what to do), execution (the reps performed: layer 3), and the **execution system** (layers 1-2 plus enforcement and gardening: the machine that makes the reps reliable and compounding). The aphorism is true because strategy is a one-sentence layer-3 utterance, cheap to produce, while the machine is expensive, slow, and rare: the machine is the moat. Better: a good machine repairs a mediocre strategy (the register turns wrong bets into accumulated learning, and learning iterates the strategy), while a brilliant strategy inside no machine never becomes rows. Precisely restated: **the execution system eats strategy, because it converts strategy into data and data into learning.**

Part 8: The lineage (ten thousand years of the three layers)

The model is not modern. Each historical crucible had a maximal product of coordination surface x cost of failure, and each added one element to the stack:

Granary -> Army -> Monastery -> Church -> Ledger -> Railroad -> Factory -> Software -> Agents

- **The granary: writing itself was invented for layer 3.** The lineage's true start is pre-cuneiform clay-token accounting (~8000 BCE, documented by Schmandt-Besserat: tokens standing for grain and livestock, sealed in clay envelopes), which evolved into cuneiform temple registers by ~3000 BCE. The oldest documents in existence are not poetry: they are registers of grain, livestock, and labor. Accounting predates literature by centuries. The state and the harvest (famine EV) produced the first registers.
- **The army: enforcement under adversarial selection.** War added what the granary lacked: your structure competes against another structure, losers are annihilated, winners get copied (a Red Queen race, which is why military ontology evolved fastest). Rome: the legion's full stack, from the contubernium to the marching camp rebuilt to an identical layout every night, drills that Josephus called bloodless battles, capital enforcement, and documented schema migrations (the Marian reforms). The payoff: average farm boys became the world's best infantry for five centuries, and **the system survived bad generals**: Rome could lose an entire army at Cannae and regenerate, because the ontology was reproducible and talent was not the load-bearing part. Genghis Khan: the decimal system (arban 10 / zuun 100 / mingghan 1,000 / tumen 10,000) used as a POLITICAL weapon: he broke tribal units and reassigned men across tribes, re-pointing loyalty from clan to structure: re-authoring the nouns to change what people are. Plus the yam relay network (declared infrastructure), merit promotion, collective-responsibility enforcement wired into the unit, and the nerge: the annual great hunt, a scheduled ceremony whose real function was drilling tumen-scale coordination. Mongol armies converged on schedule from continental distances and operated for years thousands of kilometers from Genghis: the absence test at its historical maximum. **The Mongols were the fastest force on earth BECAUSE of the structure**: the standing rebuttal to "process kills speed." The Prussian General Staff later added institutionalized gardening (after-action learning, war games) and Auftragstaktik: commander's intent plus local improvisation, structured improvisation as formal doctrine. The caution is in the same lineage: late Rome is what an ungardened ontology becomes.

- **The monastery (Rule of St. Benedict, ~530 CE): the clocked ceremony.** Canonical hours as scheduled ceremonies, defined roles, and a weekly enforcement ritual (the chapter of faults: violations surfaced aloud). It passed the absence test for fifteen hundred years across thousands of houses: history's longest-running organizational system. Mechanical clock discipline itself came out of it.
- **The church: the amendment sitting, and the record's largest gardening failure.** The monastery is one house's rule; the Church as an institution ran the full stack at civilizational scale, and what it adds to the lineage is the slow loop as a standing institution. Layer 1 is literally called canon law (major versions shipped in 1917 and 1983); sacraments are typed state transitions on a soul's lifecycle; sainthood is a register with an admission process whose examining officer's actual title was **advocatus diaboli**: adversarial review institutionalized four centuries before software engineering named it. Layer 2: the Mass on the liturgical clock, with procedural instructions literally called rubrics. Layer 3: parish registers of baptisms, marriages, and burials, among the oldest continuous records in Europe. The ecumenical councils, Nicaea to Vatican II, are amendment sittings; ordination is a recorded, scoped authoring grant (apostolic succession as a two-millennium authority register) while the laity operate freely; and the parish is a franchise unit externalized enough that an average priest can run one. The caution is what earns the entry its place: **the Reformation is the largest gardening failure on record**. Indulgences drifted over three centuries from enforcement mechanism into a billing system, and the drift's owner was the house itself; Luther filed the 95 Theses as a licensed author through the system's own doors (disputation points, in Latin, sent to his archbishop the same day); and the institution processed the amendment proposal as a discipline case. Enforcement ran where gardening should have, a practice question was escalated into an authority crisis, and the organization forked. Causality stated honestly, because survivorship cuts both ways (this entry, like the whole lineage, is illustration rather than evidence): no institution survives twenty centuries *because* it adopted this machinery; the survivors are the ones that kept evolving it, and the machinery is why their structure compounds rather than merely grows.
- **The ledger (Venice, codified by Pacioli 1494): verifiable data.** Double-entry bookkeeping is arguably the most durable ontology ever authored: unchanged in its core for five centuries and still the substrate of every finance system running today.
- **Railroad and factory: scale coordination and measurement** (the first modern org charts, timetable discipline, Taylor). **Software: executable SOPs** (the process runs itself). **Agents: free enforcement** (the checker at every write path, which history could never afford: see Part 5.5).

Ten thousand years, same three layers; the module is the newest entry in a very old lineage, and the lineage is the lecture's credibility in one slide.

Appendix: governance

The steward's block: version history, epistemic-stamp governance, and the unit-register declaration. Folded out of the reading head in every render; the source keeps it at the top for the gate.

Version: 3.4.0 (2026-08-29). Semver, one line, bumped in each amendment commit: major = vocabulary or model breaks (anything derived artifacts or trained cohorts must migrate to), minor = additive laws and sections, patch = wording. v1.0.0 is the 2026-08-20 "v1 complete" baseline. v1.1.0 (2026-08-26, mislabeled minor by its own scheme: it retired the "meta-activities" framing and rewrote the calcification caution, both breaking) added the identity law, the two loops and reification (Part 1.25), the survey door (Part 2, skill 2), the five verbs (Part 2.5), and the calcification relocation (Part 5). v2.0.0 is the adversarial-review-v4 fix pass: the new taxonomy migrated into every derived artifact, each stamped with the canon version it builds against, enforced from here on by the release gate (`scripts/release-gate.sh`; no version-bump commit without a passing run cited in its message). v2.1.0 adds the epistemic-stamp governance below and the first provisional law (spec and implementation, Part 1.25). v2.2.0 adds the two-kinds-of-processes law (Part 2.5: domain-owned SOPs versus ceremonies as cross-domain convenors, provisional with its first receipt logged). v2.3.0 (from the email build's first operating night): the survey-door guard sharpened into the four-part jobs admission standard with the retro-fit trap and late-entry clause; the flip-conditions law (Part 4); the induction law (Part 2.5). v3.0.0 (2026-08-27): the doors RENAMED to the pain door and the JTBD door (breaking; the field-tested names from the first live teach displaced incident/survey before any cohort learned them, swept everywhere); the depth and split rules on the doors (Part 2, skill 2); the construction-grounds law with the coverage walk, and the authority law (both Part 1.75, provisional); the spine defined as a term of art (Part 2.5); Palantir Foundry added to prior art (Part 5.5); the KYC compliance audit joins the worked examples. v3.1.0 (2026-08-27): the grain law (Part 2) — a committed domain is a pipeline, never a category or an outcome; the four sizing tests, the organ corollary, and the walk-capacity guard; distilled from the fleet's first sizing ruling. v3.1.1 (2026-08-27): spec-and-implementation promoted to SETTLED on its first receipt (the email-intake concern's court read-state divergence, ruled an amendment; the first full audit of a JTBD-door structure against the v3.1.0 instrument set ran the same sitting). v3.1.2 (2026-08-27): the mantra-to-model note (Part 6): guard names the fast loop whole, Inspect its human half. v3.2.0 (2026-08-28), the field harvest from Relay's first two concern-authoring days: reification, the authority law (gaining the apprenticeship grant path), and the grain law promoted SETTLED on their pre-registered receipts (the meta-concern; the live apply-door allowlist; the second independent sizing ruling); the spine-anatomy law with the orphan rule (Part 2.5); the jurisdiction-and-home law (Part 1, provisional); derivation-before-delegation (Part 5, caution 3). v3.2.1 (2026-08-28, same night): the jurisdiction law promoted SETTLED on its pre-registered receipt (the DEC-register re-homing, DEC-50, run through the six-step procedure hours after ratification), and its executor's friction note folded in: re-homing named in two shapes, custodial (stewardship and address move) and jurisdictional (only the governing spine changes; alias and handover satisfied vacuously, legally). v3.3.0 (2026-08-29): the canon reified into its own ontology — unit taxonomy (MODEL/LAW/INSTRUMENT/EXAMPLE/NOTE) + REGISTER.yaml (24 laws in 7 named families, 7 models, 7 instruments; NAMES as identity, slugs for machines, Parts demoted to reading order, prose-from-rows refused with a flip condition); the cognitive-fit law (Part 4: structures sized to the operating head — Norman's working-memory rationale, reverse-deriving the split line); the ranking corollary of the identity law (position is a field, never an identity; from the Gotrade principles-reordering question). v3.3.1 (2026-08-29): the Church joins the lineage (Part 8, between the monastery and the ledger: canonization's advocatus diaboli as institutionalized adversarial review, councils as amendment sittings, ordination as the recorded authoring grant, the Reformation as the record's largest gardening failure; causality stated as illustration, not evidence); the grain law's provenance cites its ruling code (DEC-29); the print render folds the governance header into an appendix, matching the site (derived views only, source untouched). v3.3.2 (2026-08-29): the readability edition — sixteen laws promoted to headed subsections (register name as the heading, statement lead, short paragraphs, provenance and receipts demoted to a trailing stamped line), the two remaining walls split in place (skill 2, the calcification caution); prose structure only, no vocabulary or unit change (the ranking corollary applied to the canon's own reading order); renders gain law-level anchors, stamp chips on law headings, hover anchor links, and styled provenance lines. v3.3.3 (2026-08-29): Part 7 retitled to "The honest scope" — the "(say this out loud in the lecture)" parenthetical was a stage direction to the lecturer mis-homed in a reader-addressed title; delivery's home already carries it (Part 6's lecture row names the spoken set, the body's "stated up front" line keeps the honesty reader-facing). NW-ANA-23. v3.4.0 (2026-08-29): the organ type system (Part 2.75) — eight organ types EXTRACTED from the two living deployments' inventories (register, ledger, court, board, surface, door, index, corpus; the spine excluded as the organ-holder, ceremonies as processes; scoreboard and form marked deployment-local with flip conditions); the types-and-bindings law (Laws of the spec, provisional: the canon owns types, deployments own only bindings; config-as-data source `canon/ORGAN-TYPES.yaml` carrying types only; gate check 6 alarms on count

drift); the qualifier corollary of cognitive-fit (Laws of the head, settled on the view-only mini-app ruling). One language across every deployment Norman runs; NW-ANA-22.

Epistemic stamps (ratified 2026-08-26): every law is either SETTLED (receipts cited) or PROVISIONAL (ratified, no receipts yet, its first receipt's due condition stated inline). Laws enter the canon the day they are ratified; the stamp is what keeps eager canonization honest while the philosophy actively develops. Provisional laws are taught, drilled, and examinable everywhere EXCEPT the day-60 certification verdict, which may not hang on a provisional law alone (the reversibility ladder applied to the course's own instruments: the certificate is the irreversible act). A provisional law whose receipt window lapses without a receipt is a kill-docket candidate; promotion to settled is a patch-level edit citing the receipt.

The unit register (declared 2026-08-29, v3.3.0): the canon's own ontology, applied to itself. Every unit is typed — MODEL (descriptive structure), LAW (normative, stamped), INSTRUMENT (runnable), EXAMPLE, NOTE — and registered in `canon/REGISTER.yaml`: laws by NAME (the primary identity; human memory runs on labels) and slug (the machine identity), grouped into at most seven named families of at most five laws (the cognitive-fit law obeyed by its own canon: 7 models, 7 instruments, 7 families). Part numbers are READING ORDER only, free to restructure at editions; nothing cites them as identity. The register is metadata, never prose — compiling the canon's text from rows is REFUSED with a pre-registered flip condition (a second edition needing a different ordering or subset of the same laws rendered from one source): the half-hour read is authored, not rendered.