EVIDENCE-GRADED

BUILDING AN AI-FIRST ORGANISATION

A methodology for building AI-first, small or large.

Five gated stages wrapped in a self-correcting loop. Read it by organisation type, run the stages, and use the implementation runbook to know exactly what to put in place. Synthesised from primary lab, VC and consultancy material, then adversarially fact-checked.

- **FRAME** · DECIDE WHAT TO BUILD & WHY IT COMPOUNDS
- **FOUND** · BUILD THE CONTROL PLANE FIRST
- **FORGE** · BUILD THE FIRST CLOSED LOOP
- **FLYWHEEL** · COMPOUND THE ASSET, RUN AI-FIRST
- **FEDERATE** · SCALE THE SUBSTRATE TO NEW DOMAINS

Ben Emson
benemson.com

A practitioner's field guide. Generic and shareable; every example is a public framework or a neutral illustration.

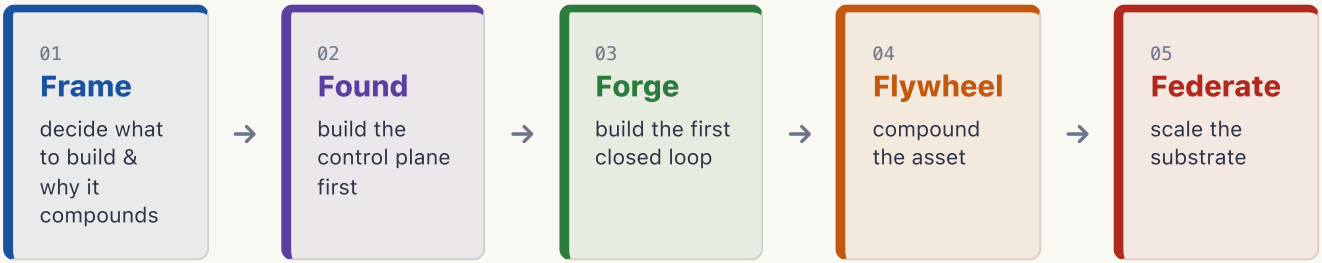
THE ONE IDEA, AND THE SHAPE

The spine

Five stages, each separated by a hard gate you cannot pass without evidence. A continuous meta-loop wraps all of them. The gates are the methodology; the stages are just where they sit.

THE THESIS

An AI-first organisation is a set of closed loops that compound a proprietary asset — data, codified workflow and evals — while spending inference instead of headcount.



◀ META-LOOP · RE-VERIFY · STRESS-TEST VS MODEL UPGRADES · RE-RUN EVALS ON LIVE TRAFFIC · PRUNE ▶

It is a work-design problem, not a tech problem

Roughly 10% of the effort is algorithms, 20% data and tech, 70% people, process and change. Treat AI as an IT project and it fails.

BCG 10-20-70 (directional)

The moat is a result of execution, not a precondition

At 0-to-1 the only thing that matters is execution. You build defensibility by deploying, not by planning it.

YC, adapting Helmer's Seven Powers

EACH STAGE, AND THE GATE THAT LETS YOU MOVE ON

The five stages

You cannot pass a gate on a promise. Each one needs evidence; any red light stops you.

STAGE 1 **Frame** decide what to build

Map an end-to-end workflow; find the segment that is high-volume, repetitive, money-adjacent and painful. Run it by hand yourself before automating it.

> **Gate:** a named user who'll pay (5+ pre-sales) · a measurable outcome · feasibility on 20–50 hand-scored inputs · a moat hypothesis.

STAGE 2 **Found** make it loop-ready

Build the control plane before any autonomy: legibility (record everything), owned memory, permissions, observability, rollback, human override, cost. Reuse a harness; don't rebuild the loop.

> **Gate:** governance + control plane exist before you orchestrate agents. Every action tagged reversible-or-not.

STAGE 3 **Forge** build the first closed loop

One workflow, fully closed-loop. Evals before code; a Planner → Generator → Evaluator harness with the judge separated from the builder; one feature at a time; test as a human would.

> **Gate:** the loop improves the Stage-1 outcome at pass^k reliability — all of k repeated trials succeed, not one lucky run.

STAGE 4 **Flywheel** compound the asset

Close the loop so it runs with minimal intervention: traces → human + LLM feedback → automated evals (a gate) → rank → implement → repeat. Build the company brain; spend tokens not headcount.

> **Gate:** the asset demonstrably compounds — measurably better this month than last, with the eval gate holding.

STAGE 5 **Federate** scale the substrate

The same substrate runs adjacent workflows and verticals; central guardrails + domain-level execution; roles compress to IC (builder/operator) and DRI. Fund expansion on adoption and outcomes.

THE SAME METHOD, READ FOR WHO YOU ARE

By organisation type

Each archetype has a binding constraint, a place to start, and a first move. Mode and tier compose with these.

Solo builder / indie

1 PERSON · GREENFIELD

Constraint: your time and focus.

Start: Frame → one Forge loop; one harness, a few skills.

```
This week: name one user, run the workflow
by hand once, write one SKILL.md.
```

Early startup

2–15 · GREENFIELD

Constraint: finding the loop before runway ends.

Start: all five stages in order; forward-deploy for first accounts.

```
This week: pre-sell to 5; if 0–1 bite,
change the idea, not the pitch.
```

Scale-up / growth

COMPANY · MATURING

Constraint: the loop calcifying as you grow.

Start: deepen the Flywheel; begin Federate to a second domain.

```
This week: stand up the nightly loop and a
real eval-gate on prod traffic.
```

SME / mid-market

NON-TECH-NATIVE · TRANSFORM

Constraint: no in-house AI muscle; risk-aversion.

Start: Transform mode on ONE beachhead; buy the harness, don't build it.

```
This week: pick one painful, measurable,
high-volume workflow; name an owner.
```

Large enterprise / incumbent

500+ · TRANSFORM

Constraint: 70% is people/process/change; legacy; governance.

Start: Found, then Forge on a GM-owned beachhead. Federated, not centralised. You do not need to re-platform the stack.

```
This week: name an exec sponsor + one
beachhead; stand up the control plane.
```

Agency / services firm

ENGAGEMENTS INTO IP

Constraint: margins; bespoke work that doesn't compound.

Start: the forward-deployed motion — embed, build evals from client data before code, productise the pattern back.

```
This week: pick one repeatable client
problem; build its eval suite first.
```

Regulated / high-stakes — health · finance · legal · safety

Constraint: reversibility, auditability, harm avoidance. **Start:** Found is non-negotiable and heavy — human gates, audit trail, all-pass (not partial-credit) evals, closed-universe pilots. Keep a human in the loop on anything that can't be undone.

```
This week: write the policy/approval gates and the all-pass eval rubric before building anything.
```

THE MINIMUM ORDER TO BUILD IN

The build sequence

Do these in order. Step 6 is a hard gate: never pass it on a single lucky run. Fit the weight to your size — solo builders run the first five and defer the rest.

1. **Frame charter**
   Named user, measurable KPI, feasibility test, moat, kill criteria.
2. **Store + tracing**
   Record every conversation and tool call from day one. Cannot be retrofitted.
3. **Harness + policy table**
   Choose a harness; tag every action reversible-or-not.
4. **Write the evals first**
   20–50 cases drawn from real failures, before any production code.
5. **Build one loop**
   Planner → Generator → Evaluator, with the judge separated.
6. **GATE · pass^k**
   The loop improves the KPI across k repeated trials. Only then proceed.
7. **Online evals + nightly loop**
   Score a sample of live traffic; propose skill edits each night.
8. **Dashboard + weekly review**
   Metrics that matter; a human reads a transcript sample weekly.
9. **Company brain + pricing**
   Queryable models over recorded data; outcome-aligned pricing.
10. **Federate**
    A second workflow on the same substrate, without re-architecture.

THE RUNBOOK — THE STANDING PARTS OF AN AI-FIRST ORG

The five layers to put in place

An AI-first organisation is these five layers, standing. The first two are foundations; the third is the steering wheel most teams skip.

L1 Control plane

BUILD FIRST · FOUNDER / PLATFORM

- single system of record
- tracing on every run + tool call
- owned memory layer
- policy / reversibility
- cost controls
- provider-agnostic harness

L2 Skill & prompt layer

BUILDERS

- skill registry (SKILL.md in git)
- prompts versioned, out of app code
- staging → prod + rollback
- versioned tool contracts (MCP)
- LLM-land vs code-land split

L3 Eval & feedback layer

THE STEERING WHEEL · AN EVALS DRI

- golden set from real failures
- capability + regression evals at pass^k
- online evals on a prod sample
- calibrated LLM-judge
- nightly improvement loop
- metrics dashboard

L4 People & cadences

LEADERSHIP

- a DRI per outcome
- a named evals owner
- exec sponsor + GM beachhead (enterprise)
- weekly transcript review
- harness stress-test per model upgrade
- assumption re-verification

L5 Decision records

WRITE ONCE, REVISIT · FOUNDER / SPONSOR

- charter (user + KPI + sample size)
- moat hypothesis
- harness + model choice
- policy / reversibility table
- the metrics you steer by
- pricing model + kill criteria

MATURITY SELF-SCORE · RATE EACH LAYER 0–5 · YOUR LOWEST LAYER IS WHAT TO PUT IN PLACE NEXT

WHY MOST ATTEMPTS DIE — AND THE GATE THAT CATCHES EACH

Failure modes

Each gate in the methodology exists because of one of these. If you skip a gate, you invite its failure.

Building before a user

Months of beautiful product for no one.

Caught by: the Frame named-user gate (5+ pre-sales or stop).

No measurable outcome

The loop can't compound; outcome pricing is impossible.

Caught by: the Frame measurable-outcome gate, or exit.

Pilot purgatory

Endless demos that never reach production.

Caught by: the Forge gate — improve the real outcome at pass^k first.

Over-automation harm

An autonomous agent makes a bad, irreversible call.

Caught by: the Found control plane — reversibility + human gates.

Vendor-locked memory

Your institutional knowledge lives in someone else's platform.

Caught by: Found — own your memory layer.

Solo over-engineering

A one-person team builds enterprise machinery, ships nothing.

Caught by: the Solo tier — minimum loop only.

Services-becomes-consulting

Bespoke client work that never productises.

Caught by: FDE discipline — merge a reusable feature back each time.

Playbook rot

The method ages as models and the field move.

Caught by: the Meta-loop — re-verify, stress-test, prune.

"It's an IT project"

Treating transformation as tech, not work-design.

Caught by: 10-20-70 — a GM owns it; 70% is people/process.

WHAT IS DIRECTIONAL, AND WHEN NOT TO USE IT

Read before copying

Synthesised from interested-party sources (labs, VCs, consultancies). Cross-source convergence is the main reason for confidence; some of it is forward-looking.

Directional, not proven — handle with care

Forward-looking: outcome-based pricing and the collaboration-layer moat are predictions, not settled results.

From enterprise cases: the 70-20-10 split and forward-deployed unit economics may not generalise.

Verified deepest: the build-loop material (eval-driven development, Planner/Generator/Evaluator) is primary and reproducible; the strategy frameworks are credible but partly marketing-adjacent.

A myth, refuted: transforming an incumbent does not require re-platforming the entire stack. Start on one workflow.

When NOT to use this: if the outcome is genuinely unmeasurable, if most actions are irreversible and can't be gated, or if there's no path to a proprietary asset — you're building a wrapper that gets commoditised.

THE FULL METHODOLOGY — FREE

The interactive version: find your path, run the gates, score your readiness.

benemson.com/resources/ai-first-business-methodology

The evidence-graded reference this field guide is drawn from: a diagnostic that tailors the method to you, every stage with its metrics and templates, and a maturity self-score.

DRAWN FROM, AMONG OTHERS

YC — The 7 Most Powerful Moats for AI Startups
Anthropic — Demystifying Evals · Harness Design
Bain — Roadmap to Reality · Next Operating Model
a16z — Big Ideas 2026
OpenAI — Agent Improvement Loop (cookbook)
BCG · McKinsey · IBM — 10-20-70 · Rewired · 7-gate