

One discipline governs agent work from intent to release

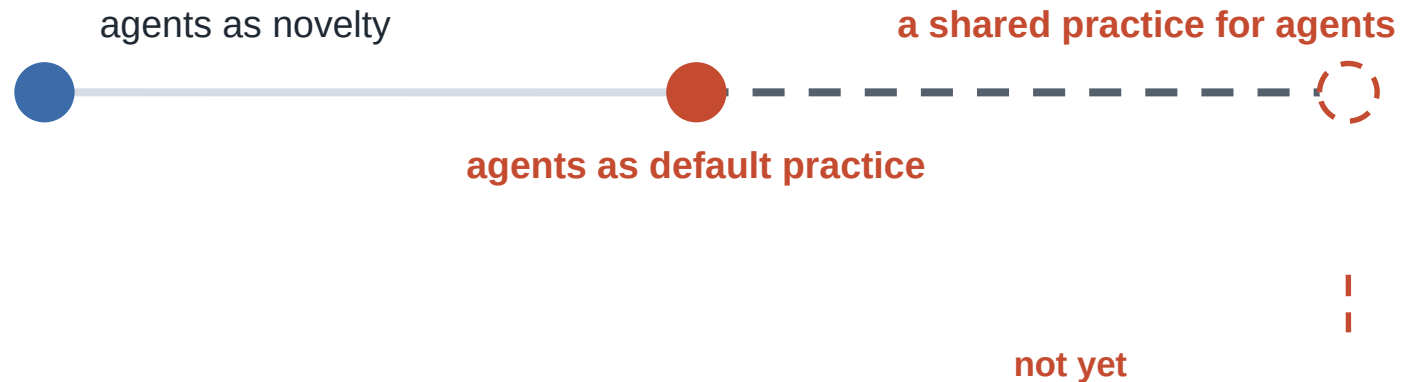
An evaluation briefing for engineering leaders whose teams build
with AI agents



September 2026

2 Most developers ship agent-touched code every day

- Most professional developers ship AI-touched code daily
- The output is locally good and globally inconsistent
- The document disciplines human teams built never reached the agents
- Every practice era gets its equivalent of test-driven development



agent-touched code became the default before any discipline caught up

3 Agent code drifts without a shared practice



Drift

specs and code
diverge between
sessions



Contradiction

a design ignores an
earlier decision

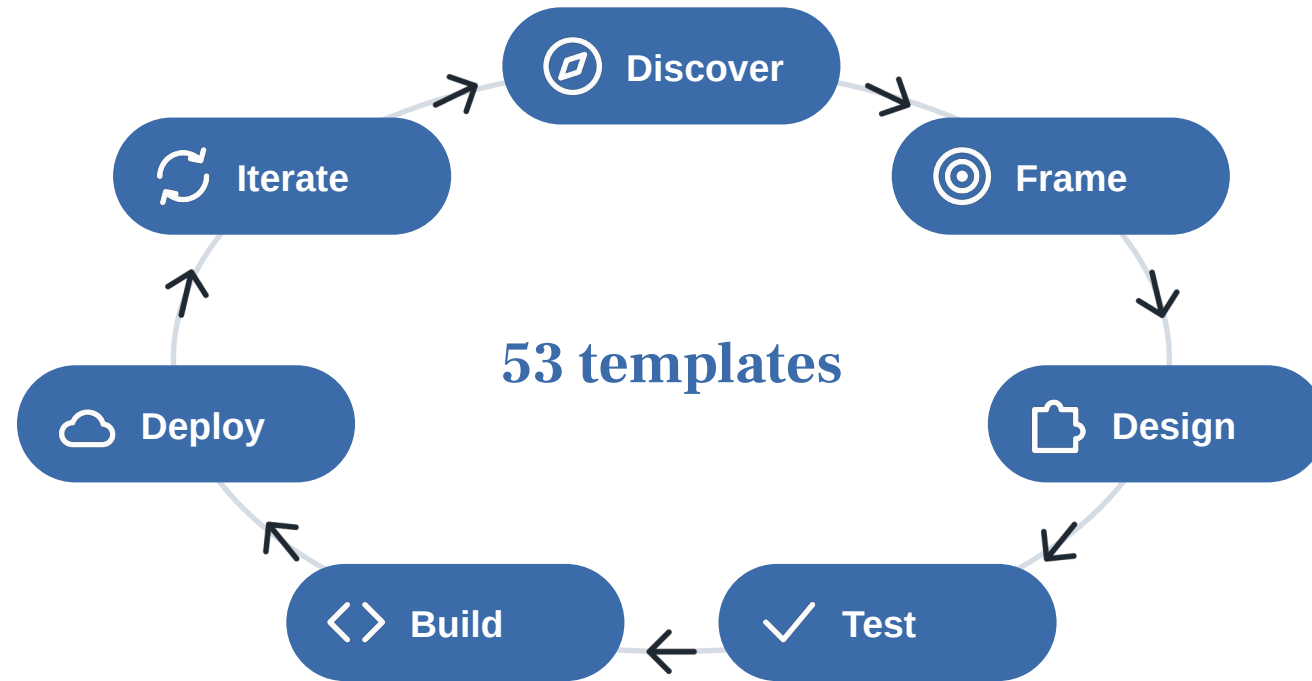


Cold start

every session starts
from zero

- Drift surfaces in review, after the work, so each catch costs a rewrite
- A contradiction with a recorded decision surfaces in production, where it costs an incident
- A cold start costs the reviews nobody remembered to run

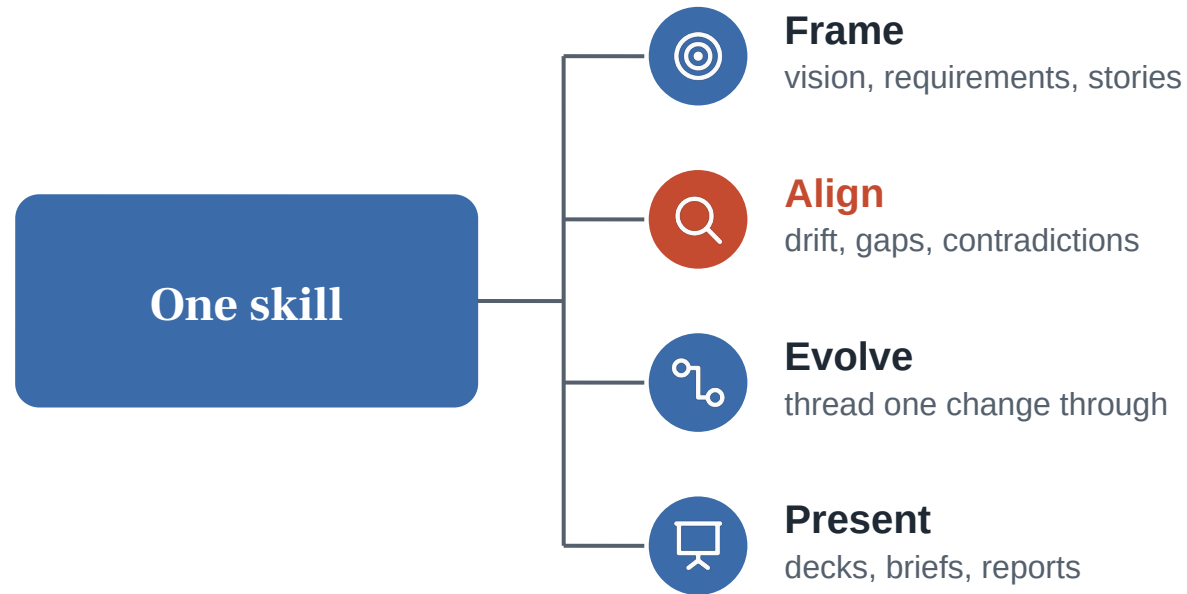
4 HELIX is the shared practice, built from templates



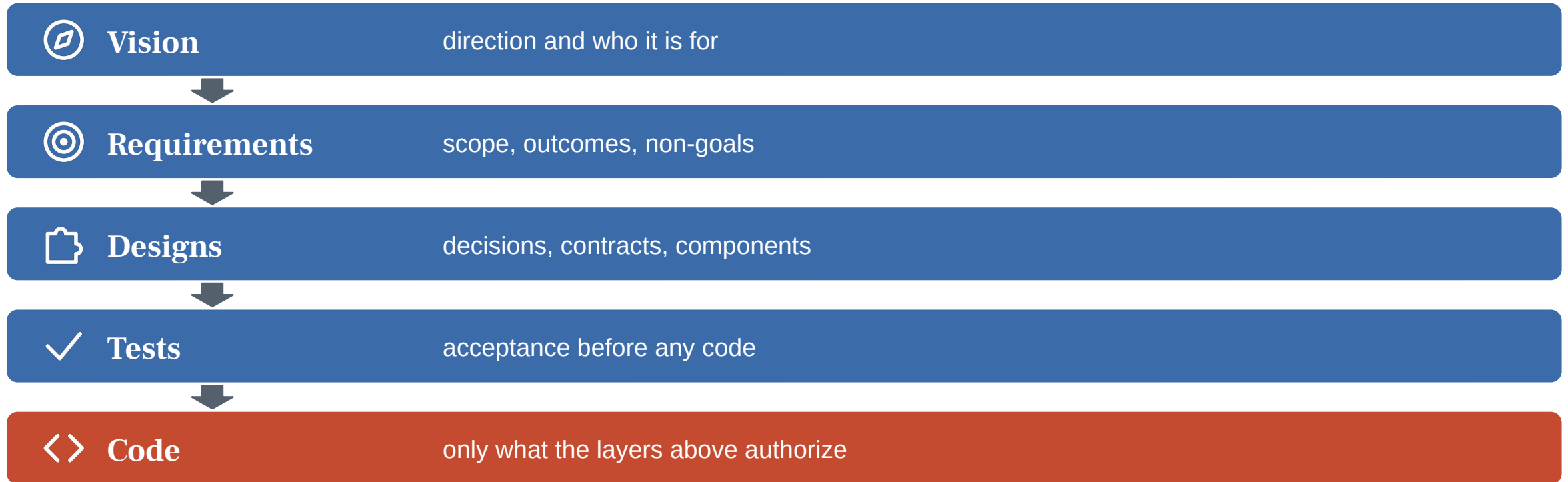
a method you adopt one template at a time, with no platform to join

5 One skill reads the templates and routes each request

- A single skill installs on the agent runtime and reads the same documents
- It routes each request to one of twenty-one jobs, from vision to deck
- The alignment check, which every team runs, finds drift and hands back a plan
- The skill body is portable: no runtime commands, so it loads on any host



6 Each template layer governs the layer below



a change enters at the top layer it touches and flows down

7 Shared choices and gates carry control across layers



Without layered control

Each agent re-decides framework, auth, and test approach per task

Drift shows up in review or in production

A claim of done goes unchecked



With layered control



The team picks each shared choice once and every agent inherits it



Gates stop work moving down a layer



Every claim cites evidence a validator can check

Verdict: the rules live in the layers, not in a prompt

8

Those same layers produce the documents sponsors read



Status report

for sponsors and clients, from the iteration plan, with evidence per claim



Deck or brief

for executives and customers, from any governed document set, every number sourced



Release notes

for users and support, from the deploy documents

9 In the worked example, one gate flags six documents

- The product vision walks one change, adding OAuth login, through the alignment check
- Six documents need work before code: 3 specs, 2 designs, the security architecture
- It also flags 1 conflicting decision, 4 missing stories, and 1 test plan
- The plan comes back in authority order: architecture first, then specs, stories, designs



3

specs

affected by the
change



2

designs

affected by the
change



1

decision

in conflict with
OAuth



4

stories

missing and
now listed

10

People decide and agents draft at every gate



People hold



What to build and which decision stands



Approval that turns a draft into authority



When the work is complete



How much runs unasked, a three-position dial



Agents hold

Drafting every document in its template

Checking each draft against the layer above

Surfacing drift, contradictions, and gaps

Stopping at the triggers and handing judgment back

Verdict: every hand-off has an owner, and HELIX ships no runtime or tracker and approves nothing

11

Pilot HELIX for one quarter and count what gates catch

Run HELIX on one project for one quarter and count the findings the team accepts as real

Step	Owner	Date
 Pick one active project with an agent already in the loop	head of platform	this week
 Adopt the templates and run one alignment check on its documents	the project's team lead	sprint one
 Review the findings the team accepts as real, and what each would have cost in review	both of us	week four

Without a trial the practice stays improvised and drift keeps surfacing in review

Sources

Id	Claim or figure	Governing artifact and section
S1	A methodology and artifact catalog for AI-assisted teams: templates plus one skill	docs/helix/00-discover/product-vision.md#mission-statement
S2	AI-assisted development went from novelty to default; most developers ship AI-touched code daily	docs/helix/00-discover/product-vision.md#why-now
S3	Three failure modes: artifact drift, inconsistent agent behavior, lossy session boundaries	docs/helix/01-frame/prd.md#problem
S4	Templates for 53 artifact types plus a single routing skill; no CLI, tracker, or execution loop	docs/helix/01-frame/prd.md#summary
S5	The seven activities, the artifact loop, and the authority hierarchy	workflows/README.md#activities, workflows/README.md#authority-hierarchy, workflows/artifact-hierarchy.md
S6	Twenty-one routed modes in the routing table; alignment is the required mode	skills/helix/SKILL.md#routing-rules, docs/helix/01-frame/prd.md#goals
S7	Four-step contract: write the brief, check alignment, plan the work, run it	docs/helix/01-frame/prd.md#core-workflow-contract

Sources (continued)

Id	Claim or figure	Governing artifact and section
S8	OAuth example: three specs and two designs affected, one decision in conflict, four stories missing	docs/helix/00-discover/product-vision.md#user-experience
S9	Concerns propagate practices downstream; the project's active concerns and slots; 50 concerns in the library	workflows/README.md#cross-cutting-context, docs/helix/01-frame/concerns.md, workflows/concerns
S10	Seven principles, from spec is the contract to deliverable over machinery, with tension resolution	workflows/principles.md#principles, docs/helix/01-frame/principles.md
S11	Artifact honesty: phantom claims block; verification evidence comes from the running system	docs/helix/01-frame/features/FEAT-016-artifact-honesty.md, workflows/concerns/verification/practices.md
S12	Tests cite the criterion they cover; quality floors never slide back	docs/helix/02-design/adr/ADR-009-acceptance-criteria-ownership.md, workflows/ratchets.md
S13	Non-goals: no CLI, tracker, queue, or engine; no imposed technology; judgment stays with people	docs/helix/01-frame/prd.md#non-goals

Sources (continued)

Id	Claim or figure	Governing artifact and section
S14	Six install surfaces: DDx, Claude Code, Codex, Copilot, Databricks Genie, Grok Build	docs/helix/01-frame/features/FEAT-013-runtime-install-coverage.md, docs/install/README.md
S15	Autonomy is a three-position spectrum with a hard-stop floor at every level	docs/helix/02-design/adr/ADR-003-autonomy-spectrum.md
S16	Roadmap, iteration plan, status report; a client cut may subset but never contradict	docs/helix/01-frame/features/FEAT-017-iteration-documentation.md
S17	Deliverables projected from governed artifacts, every number sourced, gated and inspected	workflows/modes/present.md, workflows/activities/06-iterate/artifacts/deliverable/meta.yml
S18	Targets: < 3 findings per run; 3 runtimes in 12 months; 50+ repos in 18 months	docs/helix/00-discover/product-vision.md#success-definition, docs/helix/01-frame/prd.md#success-metrics
S19	Layers are the control: each layer governs the next; a change enters at the top layer it affects; rules live in a layer, not a prompt	workflows/principles.md#layers-are-the-control
S20	The hierarchy as control loop: authority flows down, concerns cross layers, gates sit between layers, humans hold the hand-offs	workflows/artifact-hierarchy.md#the-hierarchy-is-the-control-loop

Sources (continued)

Id	Claim or figure	Governing artifact and section
S21	Humans decide, agents draft: people own intent, judgment, approval; agents own drafting, checking, surfacing, recording; every hand-off has an owner and a rule	workflows/principles.md#humans-decide-agents-draft
S22	Hand-off table: approval, autonomy level, stop triggers, escalation, surfaced judgment, and the rule behind each	workflows/README.md#human-ai-collaboration, docs/helix/00-discover/product-vision.md#key-value-propositions