

HUMAN-LED SOFTWARE ENGINEERING

I lead the system. Agents increase my leverage.

The Gateway grew out of real product work: a way to assign AI work, control what it can change, and inspect what comes back.

I set the architecture, the standards, and the release decisions.

I use agents to build and challenge the work. Their job is to extend my capacity, not take ownership of the outcome.

Adolph White Jr. · Founder, MGV

Eight decisions that shaped the system

From shared instructions to tested operating boundaries

01 OCTOBER 2025

Give shared facts a home

Moving work between conversations meant carrying context by hand. I established shared project records, specialist roles, and a clear place for assignments.

Start from the same facts.

02 JANUARY-MARCH 2026

Separate the stages of done

Consumer-app delivery made the distinctions practical: code completion, platform approval, deployment, and customer readiness each need their own evidence.

A finished step is not a finished product.

03 MARCH 4-6, 2026

Write a repeatable method

I established the master technical specification: how work restarts, where decisions are recorded, and how an agent finds the current project state.

Keep the project larger than one conversation.

04 MARCH 13-15, 2026

Make expectations checkable

I added checks for required context and review records, alongside recovery practices and visual and accessibility standards.

Check the work, not just the promise.

05 APRIL 2026

Compare before committing

For DNA Health Insights, I commissioned competing development proposals, chose native development, and retained Replit for supporting work.

Agents challenge. People decide.

06 APRIL-MAY 2026

Make the method reusable

I brought task definition, source checks, assignments, reviews, and results into repeatable instructions that could be used across projects.

Reuse the method, not stale assumptions.

07 JULY-AUGUST 2026

Rebuild the foundation

I audited the proof-of-concept Gateway against enterprise-grade software standards and directed a ground-up rebuild, with scale and portability as design goals.

Give each responsibility one owner.

08 SEPTEMBER 2026

Prove each capability

Live tests demonstrated bounded code edits, a reviewed Codex local-commit journey, model-specific advisory routes, and accepted-version recovery.

Permission follows demonstrated capability.

A CONTROL POINT FOR MANAGED AI WORK

The assignment sets the limits. The result still has to earn acceptance.

HUMAN AUTHORITY

Set the outcome and decide what ships

The owner defines scope and acceptance. Release authority stays with people and may be granted in advance.

ORCHESTRATION

Prepare the job and evaluate the result

The orchestrator checks current context and agent availability, assigns work, and weighs findings against evidence.

01 / REQUEST

Name the job

A concrete outcome, the relevant sources, and the work that is allowed.

- A Advice or a change?**
Different effects require different routes.
- B Which agent?**
Use its qualified interface and model.
- C What proves it works?**
Define the acceptance criteria.

02 / GATEWAY & AGENT

Use the permitted route

Read-only review

The agent can inspect declared sources. Project and source writes are restricted, and the recorded route preserves output and a terminal run record.

Explicit Open Review is a separate, admitted option without that durable record.

Bounded code change

A qualified agent edits within the permitted scope. The product workflow checks the result and pauses for its required review.

Codex: at most one authorized local commit, created by the adapter. Cline/Goose: qualified exact-file editing; permissions differ.

03 / RESULT & REVIEW

Inspect what happened

- A Preserve the result**
Output, changes, tests, and code identity as applicable.
- B Challenge the work**
Assigned reviewers look for faults and support findings with evidence.
- C Accept or correct**
The authorized owner resolves the next step. A completed run is not release approval.

Recovery continues from facts.

Keep useful output. Check whether a change already occurred. Recover the missing step within its authority instead of blindly repeating the whole job.

Inside the software

LangGraph owns the supported workflow transitions. The Gateway invokes the agent, applies route-specific restrictions, and records execution. Git identifies the code version.

Outside a coding permission

Push, pull-request creation, merge, deployment, and release remain separate authorized actions. Interactive Slack correspondence is a separate coordination route.

A REAL, DISPOSABLE TEST REPOSITORY

A code change. A test. A review of the exact result.

This demonstrates the local coding workflow. It is not a customer production release.

- 1

Limit the assignment

The request named one existing file, `order_summary.py`, a specific starting revision, and permission for one local commit.
- 2

Let Codex make the edit

The native Codex agent changed the permitted file. The recorded result stayed within the allowed paths.
- 3

Check and record the change

The adapter created the authorized local commit. The acceptance test ran against that resulting version.
- 4

Review the actual result

Cline performed an independent read-only review. The Dispatch Lead's acceptance was bound to the same result before the workflow resumed.
- 5

Finish the local assignment

One resume completed the reviewed local patch. The agent's work was not replayed. Publishing or deploying would be a separate decision.

OBSERVED RESULT

One bounded edit.
One local commit.
A reviewed completion.

The record links the request, changed paths, tests, resulting code revision, and review.

Evidence: A40-R1 terminal receipt, September 8. Local-patch workflow complete; broader qualification remained open.

WHY THAT DISTINCTION MATTERS

“Finished” is not enough.

Did the process end?

That describes execution.

Did it produce the requested change?

The difference and acceptance test answer that.

Can the work advance?

Review and the authorized owner decide.

Adversarial review is an assignment to find faults.

Reviewers challenge the implementation, its assumptions, and unnecessary restrictions. Criticism needs evidence. The lead orchestrator resolves conflicting findings; people retain final release authority.

CUSTOM SOFTWARE · SYSTEM INTEGRATION · GOVERNED AI

More capacity to build. A clear owner for the outcome.

Keep the next job moving

Shared project context and recorded results give the next assignment a starting point. The team can recover missing steps without throwing away useful work.

Find problems before they travel

Limited assignments, focused tests, and evidence-backed adversarial review make it easier to identify what needs correction before work advances.

Know what can be recovered

Accepted-version records and a tested rollback path give operations a specific version to return to when a Gateway release needs to be reversed.

Choose an agent for demonstrated ability

Capability is qualified by route, model, interface, and environment. A new agent name does not silently inherit another agent's permissions.

See one job, from request to reviewed result.

Ask for a walkthrough of the bounded assignment, test, review, and recovery records.

The aim: less repeated work and a faster path to peer-reviewed working software.
No measured speed, cost saving, certification, or customer-deployment claim is made by this guide.

AVAILABLE AS AN MGV DELIVERY CAPABILITY

Engineering leadership, supported by governed agents

MGV can use this operating model in scoped software and integration engagements: define the outcome, build a useful increment, test it, review it, and hand it over against agreed acceptance criteria.

WHAT THE EVIDENCE SUPPORTS

Working, specific capabilities

Recorded advisory work, qualified bounded code edits, the demonstrated Codex local-commit journey, model-specific routing, and accepted-version recovery.

Current selection was checked September 14 against release records. Historical tests were inspected, not rerun for this guide.

FOR A DIFFERENT ORGANIZATION

Validate the environment before rollout

The Gateway is not presented here as a turnkey enterprise product. Another organization's agents, access rules, infrastructure, and release process require integration and qualification. Linux and multi-tenant operation are not yet qualified.

info@mastermindgroupventures.com