# How I work

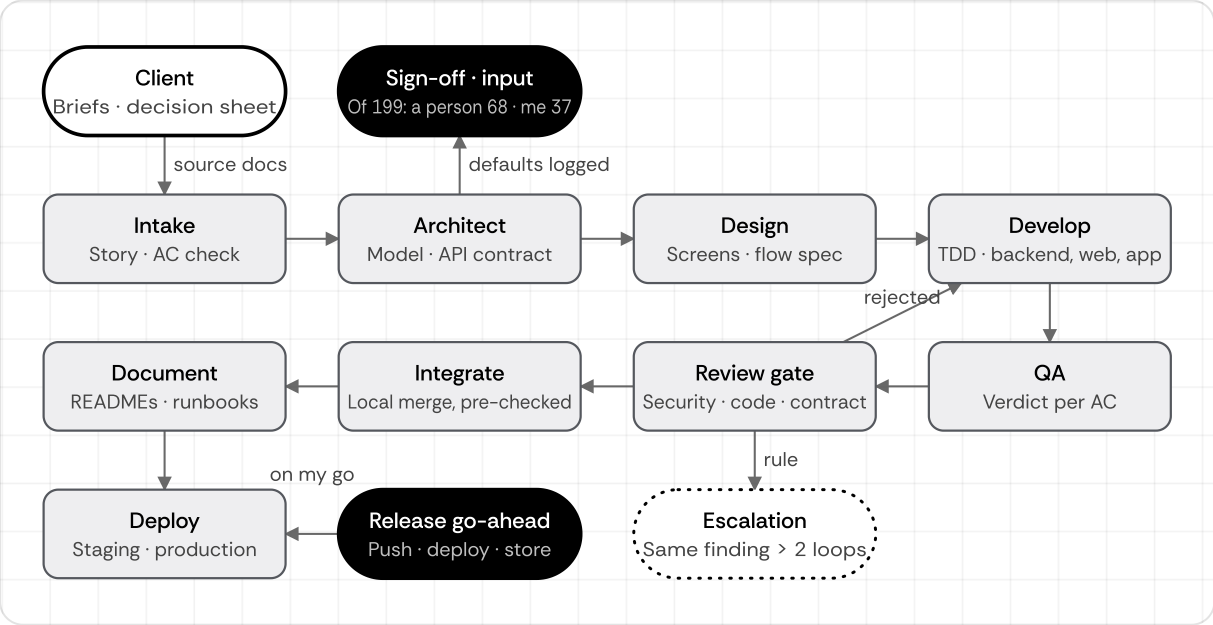
Jake Tae-moon Jo · Engineering lead · https://jake-jo-portfolio.vercel.app/en/process/

This is how I set up and ran the work on the 2026 EWA platform, and how that setup changed across six 2026 projects. Every change request from the client was shipped within 24 hours. I defined an 8-stage pipeline and assigned design, implementation, QA and review to a 13-role AI agent team. The client's stakeholders supplied the requirements and the decision-sheet answers; I confirmed or gave input on decisions taken by default, and gave the go for production.

## Pipeline

One feature, from spec to production

The eight stages in the middle are assigned to role agents. At the top the client supplies requirements and I confirm or give input on decisions taken by default; at the bottom I give the go for production.



### Client

Features come from five documents by the client's stakeholders, merged into the spec that the story IDs come from. The answers in the stakeholders' decision sheet of 3 July went in together as decisions D-128 to D-131 on 4–5 July.

### Intake

Checks the story ID and its acceptance criteria. The rule is that every feature traces to a story ID in the spec; 61.4% of the 1,855 non-merge commits carry one.

### Architect

Data model, API contract and ADRs. When a choice is needed, it goes ahead on a recommended default so work never blocks, and logs it in the decision register.

### Sign-off · input

68 of the 199 rows carry a person's confirmation or input, 37 of them with my name. 3 of the 11 hard-to-reverse ones were confirmed when logged; the rest follow the default-now, review-later mode. 26 earlier rows were reversed by later ones (some the same day), and a row that reverses an earlier one names it.

### Design

Screen and flow specs per story. Some rows reversed an earlier decision because of the design file (D-141, D-156).

### Develop

The rule is to write the failing test first; 86.4% of the 398 non-merge fix commits also changed a test file. Web types are generated from the OpenAPI snapshot, and the app's models are written by hand (D-009).

### QA

A pass or fail verdict for each acceptance criterion; a failure goes back to the responsible developer with the repro.

### Review gate

Three reviewers are in the pipeline: security when money, auth or personal data is touched, code always, and API contract when the contract or a client changes. An unresolved blocker or major finding, or any CRITICAL or HIGH security finding, fails the gate and goes back to development. Since 28 Sep, when records begin, they ran 24 times (code 12, security 10, contract 2); the spec coverage auditor sits outside the pipeline and ran 0 times.

### Escalation

If a review-and-fix loop repeats more than twice on the same finding, it stops and comes to me. This is a rule in the orchestrator's instructions; no case is on record.

### Integrate

Merges happen locally, without PRs (275 merge commits). The merge order is agreed first and conflicts are checked with git merge-tree.

### Document

Updates READMEs, runbooks and the decision register. Rules such as "a row that reverses an earlier decision says which one" also end up in the README.

### Release go-ahead

Pushing, production deploys and store submission run on my go; that is the rule. On staging, any delete-like command makes a hook ask for confirmation first.

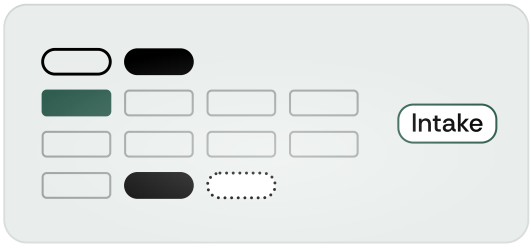
### Deploy

On my go, an agent session rolls the change out to production. Deploys are manual (ship a source archive, rebuild on the server).

## Principles

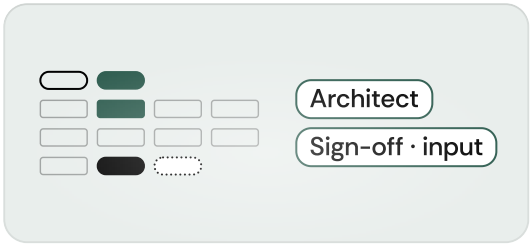
### Story-ID tracing

The rule: every feature traces to a story ID in the spec, and branches, commits and PRs carry the ID. In practice 61.4% of the 1,855 non-merge commits carry a story ID, and 70.8% carry a story ID or a decision number. There were no PRs, and 15 of the 200 branch names contain a story ID.



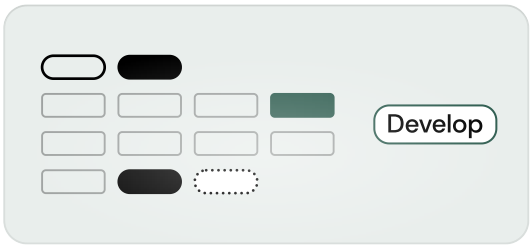
## Decision register

Instead of blocking, go ahead on a recommended default, log it, and review the defaults later in one pass. 192 of the 199 rows say how hard the decision is to reverse, and 11 are hard, locking in data or contracts. Of those, 3 were confirmed by a person when logged and 8 went ahead on the default. A row that reverses an earlier decision names it.



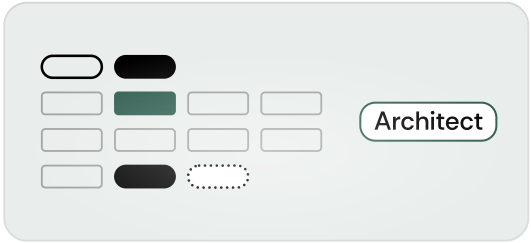

## Test first

The rule: write the failing test first. 86.4% of the 398 non-merge fix commits and 87.7% of the 764 feat commits also changed a test file. That includes edits to existing tests, so it does not mean a regression test was added each time.



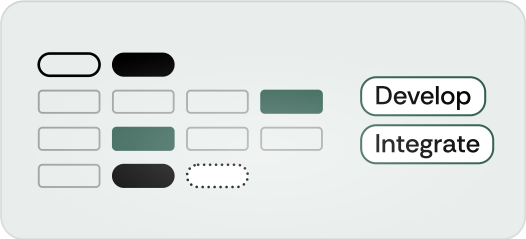

## Contract first

The one contract between repos is the backend's OpenAPI snapshot. The web generates its types from it; the app's models are written by hand (D-009, code generation deferred). When the contract changes, the API contract check looks for drift between the web, the app and the backend.



### Parallel sessions

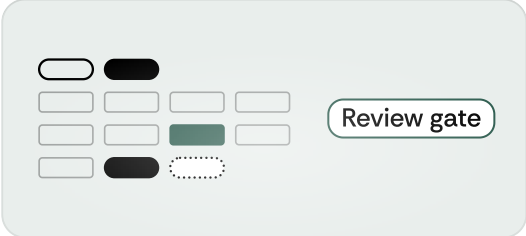
Several agent sessions run at once. Each session uses its own worktree and announces the files, decision numbers and migration names it will touch. A decision number is taken as max + 1 at the moment of writing.




## Quality gates

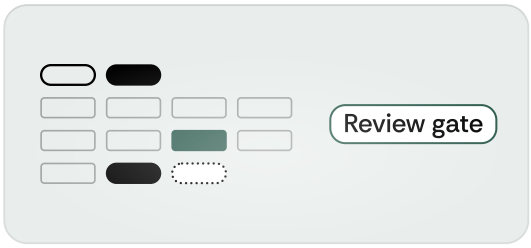
### Security review

An adversarial review of any change that touches withdrawals, settlement, payouts, auth, permissions or personal data; it does not edit code. Since 28 Sep, when records begin, it ran 10 times. 30 fix commits mention security review or a security fix (Jun 14 – Oct 29).



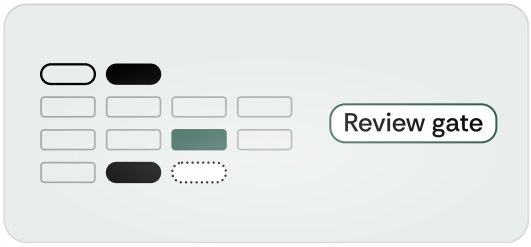
### Code review

Correctness, repo conventions, simplicity and test quality on every change. Findings are graded blocker, major, minor or nit, and the verdict is APPROVE or REQUEST-CHANGES. It ran 12 times since 28 Sep.



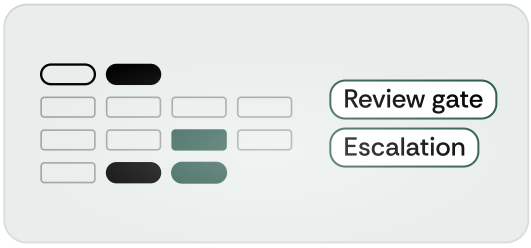

### API contract check

When the contract or a client changes, it checks that the web and the app match the backend's OpenAPI. It ran twice since 28 Sep.




### Rejection and escalation

An unresolved blocker or major finding, or any CRITICAL or HIGH security finding, counts as a failed gate and goes back to development. A further rule sends a review-and-fix loop to me once it repeats more than twice on the same finding. Both are rules in the orchestrator's instructions, and no escalation is on record.



### Guardrails

Permission settings block force-push, reset --hard and rm -rf, and three of the four repos also block reading .env. A hook asks for confirmation before every delete-like command sent to the staging server.




## How the setup changed, project by project

| Project | Agent roles | What enforced the gates | Last gate, held by | Changed from the row above |
|---|---|---|---|---|
| Apr – Jul 2026 **Construction site management** | 11 · 1 reviewer | Agent instructions (test first · 80% · commit after review and QA) | Me Merged 106 PRs | Start: 7 roles, grown to 11 |
| May – Nov 2026 **Appliance rental and maintenance** | 11 · 1 reviewer (copied) | CI (types · tests · build) · deploy after CI · typed phrase to delete | Me Merged 139 PRs<br>Me Ran the pre-go-live reset | Team copied; added CI on every PR and deploy once it passes |
| Jun 2026 – present **Earned wage access fintech** | 13 = 9 + 4 reviewers | Instructions · deny list · delete-command check hook | Me Push · deploy · store go-ahead | Reviewers 1 → 4 · check hook · reversibility per decision |
| Aug – Sep 2026 **Power market price analysis** | 1 analyst agent · 2 review passes | Stops at first failure · 21 gate checks · publish stamp | Me Adopt or revert hypotheses | Rules moved into code checks · one analyst instead of a role team |
| Sep – Oct 2026 **Screenplay translation** | 11 crew roles · a skill conducts | Instructions · tool exit codes · gates tested on broken copies | Me 3 direction calls<br>Client Per-film checks pending | Back to an 11-role team · gates test themselves |

| Project | Agent roles | What enforced the gates | Last gate, held by | Changed from the row above |
|---|---|---|---|---|
| Oct 2026 (20 days)<br>Social-graph trading | 13, EWA's role names | Instructions · deny list (same as EWA) · one sensitive change at a time | Me Check per bundle · commit on request<br>Client Answered the planning questions | Deny list back · work in bundles |

# Contact

whxoans@gmail.com
+82 10-9233-7440