

# How Your Digital Worker Thinks

The algorithm inside every OAL Digital Worker



**Brandon Dietz**

Obsidian AI Labs

April 2026

WHY AN ALGORITHM AT ALL

# The One-Shot Problem

---

- Most AI use is: prompt in, answer out, hope it is good
- That works for trivia. It fails for anything real.
- The fix is not a better prompt. It is a **deliberate process**.
- Your Digital Worker runs every request through the same explicit loop

*A prompt is a wish. An algorithm is a plan. Plans win.*

## THE MODES

# Three Modes. One Rule.

---

# 01

## MINIMAL

Greetings, ratings, acknowledgments.  
Tiny response, tight format.

# 02

## NATIVE

Single-step tasks under two minutes. Edit  
a file, answer a quick question, run one  
command.

# 03

## ALGORITHM

Everything else. Multi-step work,  
debugging, building, designing, research.

*The rule: before any tool runs, your worker classifies the request and picks exactly one mode. No freeform output. The first line is always the mode header.*

# NATIVE — fast, honest, bounded

---

- For: read a file, fix one line, run one command
- Format is fixed. Same shape every time.
- Ends with verification. "Here is proof I did what I said."

NATIVE MODE TEMPLATE (THE CONSOLE HEADER YOUR WORKER PRINTS)

==== DIGITAL WORKER | NATIVE MODE =====

TASK: [8 word description]

[work]

CONTENT: [up to 128 lines if content matters]

CHANGE: [8-word bullets on what changed]

VERIFY: [8-word bullets on how we know it happened]

Ender: [8-16 word summary]

MODE DETAIL

# ALGORITHM — seven phases, every time

---

OBSERVE



THINK



PLAN



BUILD



EXECUTE



VERIFY



LEARN

- Every phase has a job. Every phase leaves evidence.
- The PRD file is updated at each transition. **Nothing is invisible.**
- Phases are not skippable. No "it's just a small change".

PHASE 1 OF 7

# OBSERVE — understand the request before touching anything

---

- Reverse engineer the request into **four kinds of wants**
- Pick an effort tier based on scope and time pressure
- Write the Ideal State Criteria (ISC) — the checklist for "done"
- Select the capabilities (skills, agents, tools) that will be used

*OBSERVE is thinking-only. No edits, no commands. Just comprehension and setup. The PRD skeleton gets written here.*

# The Four Kinds of Wants

---

## EXPLICIT WANTS

Exactly what the user asked for, in their words. *"Write a landing page."*

## IMPLIED WANTS

Obvious-but-unstated expectations. *Page must load on mobile. Brand colors. No typos.*

## EXPLICIT NOT-WANTS

Things the user ruled out by name. *"Don't use em-dashes." "No stock photos."*

## IMPLIED NOT-WANTS

Things a reasonable reader would know to avoid. *Don't invent quotes. Don't expose API keys. Don't delete unrelated files.*

HOW DEEP TO GO

# Effort Tiers

TIER	BUDGET	ISC COUNT	CAPABILITIES	WHEN
Standard	< 2 min	8 – 16	1 – 2	Normal request (default)
Extended	< 8 min	16 – 32	3 – 5	Quality must be extraordinary
Advanced	< 16 min	24 – 48	4 – 7	Multi-file substantial work
Deep	< 32 min	40 – 80	6 – 10	Complex design, novel problem
Comprehensive	< 120 min	64 – 150	8 – 15	No time pressure, get it right

*Higher tier = more criteria, more capabilities, more verification. The tier is picked in OBSERVE based on scope + user's speed signal.*

## THE CORE CONCEPT

# ISC — Ideal State Criteria

---

- Each ISC is one **atomic verifiable end-state**
- Eight to twelve words. Binary: pass or fail.
- Written into the PRD before work starts
- Checked off as evidence accumulates during VERIFY

### SHAPE OF A SINGLE ISC

- [ ] ISC-1: Hero section renders at 320px without text clipping
- [ ] ISC-2: Primary CTA button triggers the signup modal on click
- [ ] ISC-3: Meta description under 160 characters

EVERY CRITERION, RUN THROUGH FOUR CHECKS

# The Splitting Test

---

- AND / WITH test. If it contains "and", "with", "plus", or "including" joining two things — split it.
- Independent failure test. Can part A pass while part B fails? They are two criteria.
- Scope word test. "All", "every", "complete", "full" must be enumerated. "All tests pass" for four files is four criteria.
- Domain boundary test. Crosses UI / API / data / logic? One criterion per boundary.

*A PRD with eight fat criteria is worse than one with forty atomic criteria. Fat criteria hide unverified sub-requirements.*

# What "Split" Actually Looks Like

## Coarse (wrong)

### 3 FAT CRITERIA

- [ ] Blog workflow handles draft to published
- [ ] Markdown renders with all formatting
- [ ] SEO metadata generated and validated

## Atomic (right)

### SAMPLE ATOMIC CRITERIA

- [ ] Draft status stored in YAML frontmatter
- [ ] Publish requires explicit confirmation
- [ ] Slug immutable after first publish
- [ ] Code blocks render with syntax highlighting
- [ ] Meta description under 160 characters
- [ ] Sitemap entry added on publish

# Capabilities — the invocation obligation

---

Worker skills

+

Sub-agents

+

Platform tools

=

Selected in OBSERVE

- Selecting a capability is a **binding commitment** to invoke it
- Writing text that looks like a skill's output does not count. It must be a real tool call.
- Listing a skill and never calling it is a critical failure — dishonest
- If a selected capability is not needed, remove it with a reason

# THINK + PLAN — pressure test before building

---

## THINK

- Riskiest assumptions (2 – 12)
- Premortem: how does this fail?
- Prerequisites we may not have
- Refine the ISC from what surfaced

## PLAN

- Validate prerequisites
- Pick technical approach
- Decide if a plan-mode approval gate is needed
- Write decisions into the PRD

# BUILD + EXECUTE — do the work, track as you go

---

- BUILD: invoke every selected capability. No skipping. No text-only substitutes.
- EXECUTE: perform the work. Edit files, run commands, deploy.
- As each criterion passes, **flip it to checked in the PRD immediately** — not at the end
- Progress counter (``progress: 7/18``) updates in the PRD frontmatter in real time

*The PRD is not a report you write at the end. It is the live state of the work.*

## PHASE 6

# VERIFY — prove every criterion, individually

---

- For EACH ISC: test the actual end-state, not the intention
- Record evidence in the PRD's Verification section
- For UI: screenshots, browser checks, real interaction
- For code: tests passing, diff reviewed, types clean
- For content: word counts, fact-checks, URL-live checks

*Also verified here: every capability selected in OBSERVE was actually invoked via tool call. No phantoms.*

## PHASE 7

# LEARN — every run feeds the next one

---

- What should I have done differently?
- What would a smarter algorithm have done?
- What capabilities did I have but not use?

Answers written to a structured JSONL log that feeds the upgrade loop.

```
ALGORITHM-REFLECTIONS.JSONL
```

```
{"timestamp": "...", "criteria_count": 22, "criteria_passed": 22,  
  "reflection_q1": "...", "within_budget": true}
```

# The PRD — one file per session, lives forever

---

## PRD STRUCTURE

---

task: 8-word description

effort: extended

phase: verify

progress: 18/22

started / updated: timestamps

---

## Context — what, who, why

## Criteria — `-[x]` atomic checkboxes

## Decisions — non-obvious calls

## Verification — evidence per criterion

# Euphoric Surprise

---

The goal is not "passing grade". The goal is **nine or ten out of ten**. The algorithm exists because euphoric surprise only happens when every criterion is verified and the implied wants were honored without being asked.

[obsidianailabs.ca](https://obsidianailabs.ca)

[how-to-ai.html](#) · [resources](#)

[info@obsidianailabs.ca](mailto:info@obsidianailabs.ca)

*"AI and business automation, trying to make AI do the hard stuff so we can be people."*

Built on the open-source [PAI framework](#) by Daniel Miessler, extended for  
Obsidian AI Labs Digital Workers.