REFLECTIVE AI PRODUCT

Work Alignment Guide

A calmer way to evaluate work

A live conversational AI experience that helps people reflect on what they need from work, build a clearer picture of alignment and evaluate opportunities through that context rather than treating the job posting in isolation.

ROLE

Product strategy, conversational design, AI architecture, behavioral testing

STACK

WordPress, JavaScript, PHP, OpenAI Responses API, GPT-5.5

STATUS

Live Product

Reflection → Calibration → Working Profile → Opportunity → Alignment Evaluation → Next Step

The Problem

Most job-search tools focus on the opportunity: the salary, responsibilities, qualifications and whether someone appears to match the role.

A job can look right on paper and still be wrong for the life someone is trying to build. Work shapes energy, boundaries, relationships and wellbeing.

The harder question is often not simply:

Can I do this job?

It is:

Would this way of working actually fit me?

I wanted to create an experience that helps someone understand what they need from work before evaluating the next opportunity.

> **DESIGN GOAL**
> Help people evaluate work through the context of the life they want to build, without telling them what decision to make.

CASE STUDY

How the Idea Evolved

The project began as a custom GPT for identifying potential concerns in job postings. As I worked on the experience, I realized that evaluating the job first was starting in the wrong place.

The same role can feel sustainable to one person and exhausting to another. Before the system could meaningfully evaluate an opportunity, it needed enough context to understand what the individual actually wanted from work.

That changed the product from a job scanner into a reflective work-alignment experience.

Early development focused on conversation design: helping the system listen and reflect what it was learning and gradually build enough context to support a useful evaluation. Testing different user behaviors exposed an important challenge. Some people provide rich context naturally, while others are concise, uncertain or arrive ready to evaluate a job immediately.

The system therefore could not depend on one rigid conversational path.

Development shifted toward preserving the same underlying purpose while allowing the conversation to adapt to different levels of context, confidence and participation.

Job Scan → Reflection → Individual Context → Work Alignment → Opportunity Evaluation

That progression became the foundation for the experience I ultimately built.

Key Design Decision: Understand the person before evaluating the opportunity.

CASE STUDY

Designing the Conversation

The finished experience does not begin by scoring a job posting. It begins with conversation.

Work Alignment Guide helps the user reflect on how they work best, what has felt sustainable or unsustainable in the past and what they want to protect or change going forward.

The experience moves through three broad functions:

Conversational Functions

REFLECT

Explore the user's experiences, preferences and relationship with work before drawing conclusions.

UNDERSTAND

Build a working picture of the conditions, patterns and tradeoffs that appear most important to that individual.

EVALUATE

Use that context to examine an opportunity for alignment, tension and areas that may need more information.

The stages are connected, but the conversation is not a fixed questionnaire. The amount of reflection needed can vary depending on what the user brings into the conversation.

CASE STUDY

Designing for Different Users

01

READY TO MOVE

Some users already have a strong sense of what matters to them and need less exploration before moving forward.

02

STILL EXPLORING

Others need more reflection before the important patterns become clear enough to support an evaluation.

03

LOW CONTEXT

Some users answer briefly or provide limited detail, requiring the experience to remain useful without forcing artificial depth.

04

RETURNING

A returning user may begin with prior context, but that context must remain open to revision as circumstances or priorities change.

The conversation can take different paths depending on what the user brings, while still moving toward the same underlying purpose.

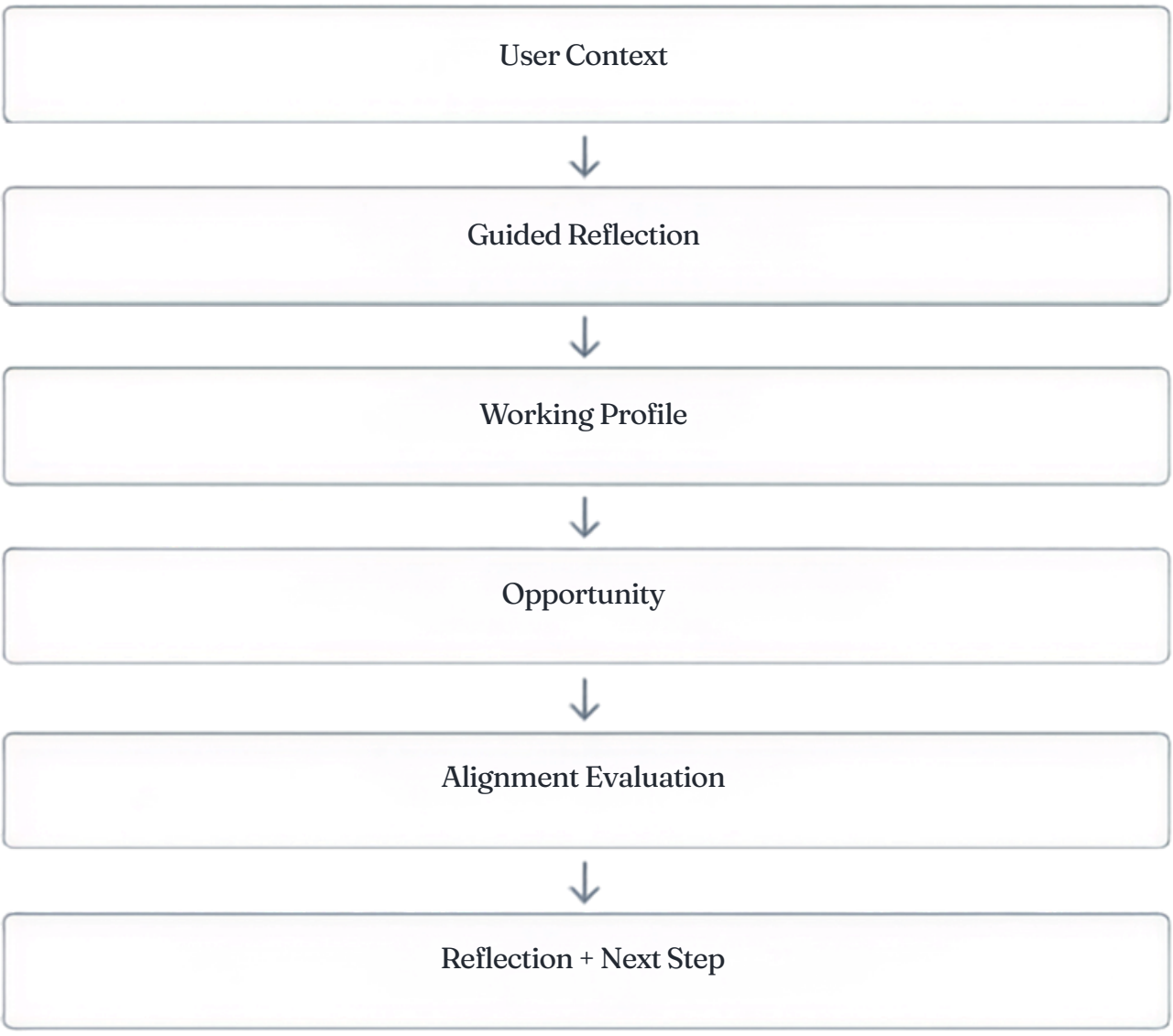
BEHAVIORAL PRINCIPLE

Adapt the path without changing the purpose.

SYSTEM
ARCHITECTURE

How the Experience Works

Work Alignment Guide connects reflection and opportunity evaluation by building a working picture of what matters to the user, then applying that context to examine an opportunity more meaningfully.



The conversation can vary along the way, but the underlying progression remains consistent: understand first, evaluate second.

Finding the Right Moment to Move Forward

One of the hardest problems was not what Work Alignment Guide should say. It was knowing when the conversation had enough context to move from reflection into evaluation.

Moving too early could produce an evaluation based on an incomplete understanding of the user. Staying in reflection too long could add friction after enough context had already emerged.

Testing showed that this boundary was more sensitive than the individual reflection or evaluation behaviors themselves. Different users reached sufficient context differently, which made one rigid conversational sequence a poor fit.

I focused testing on three possible outcomes:

1. **TOO EARLY**
   Evaluation begins before enough context has emerged, making the result less personal or useful
2. **CALIBRATED**
   Enough context has emerged to support an evaluation grounded in what matters to the user.
3. **TOO LATE**
   Reflection continues after sufficient context exists, adding friction instead of improving the evaluation.

The goal was not to make every conversation transition at the same moment. It was to make the transition appropriate to the conversation.

Key Design Insight: Consistency does not require identical conversations. It requires preserving the behavior that matters.

Testing Behavior, Not Scripts

Conversational AI will not produce the same dialogue every time, so I needed a testing standard that separated harmless variation from meaningful regression.

I built a regression suite around representative user behaviors and replayed those scenarios as the system evolved.

The tests looked beyond whether an individual response sounded good. They evaluated whether the complete conversation preserved important functions:

- reflective, collaborative tone
- appropriate pacing and user participation
- sufficient context before evaluation
- personalized synthesis
- opportunity evaluation grounded in the user's context
- appropriate handling of uncertainty
- stable closure
- responsible reuse and revision of prior context

A test could therefore use different wording, questions or turn counts and still pass if the intended conversational behavior remained intact.

TESTING PRINCIPLE

The transcript can change. The function should not.

From Prototype to Native Product

Once the conversational experience was stable in the custom GPT, the next challenge was moving it into a standalone product without changing the behavior that had already been developed and tested.

I rebuilt the experience as a native AI web application with its own interface, frontend request flow and server-side API integration.

The production architecture separates the major responsibilities:

Interface → Frontend → Backend → Product Instructions → AI Model → Response

The frontend manages the conversation experience and sends requests to a custom PHP backend. The backend validates incoming data, manages conversation history, loads the product instructions and communicates with the model through the OpenAI Responses API.

API credentials remain server-side rather than being exposed to the browser, while technical errors are handled behind the interface instead of displaying internal system details to the user.

I validated the backend independently before connecting it to the frontend, then tested the complete experience through the live website.

The objective was not simply to make the API respond. It was to confirm that the product still behaved like Work Alignment Guide after the architecture around it changed.

Key Implementation Decision: Preserve the product behavior while changing the infrastructure underneath it.

Hardening the Live Experience

A successful end-to-end test proved that the native application worked, but more testing was needed.

I reviewed the request lifecycle for places where unexpected input, network problems or repeated user actions could create confusing or unreliable behavior.

On the frontend, I added safeguards around duplicate submissions, waiting states, keyboard behavior, input handling, scrolling, timeouts and network failures.

On the backend, requests are validated before they reach the model. The system checks the request structure, message content, conversation history and required configuration while enforcing limits on message and history size.

Failures are also separated by audience. Technical details are logged server-side for diagnosis, while the user receives a simple error message rather than internal API or application information.

Later testing exposed a conversation-history issue in which the newest message could be included twice. I corrected the responsibility boundary so the backend became authoritative for adding the current message, then regression-tested the live application again.

Key Production Lesson: A successful path is only the beginning. The surrounding system has to behave predictably when something goes wrong.

When a Newer Model Is Not the Better Choice

When GPT-5.6 became available, I tested it as a potential replacement for the model running Work Alignment Guide.

I kept the established instruction system unchanged and ran the same behavioral regression suite against the newer model.

WHAT IMPROVED

GPT-5.6 produced deeper job analysis, stronger personalized synthesis and more natural alignment reasoning.

WHAT REGRESSED

Across the full experience, testing produced 5 clean passes, 2 passes with watch and 3 failures.

The failures mattered: one user pattern moved into evaluation too early, another stayed in reflection too long and returning-user behavior sometimes missed information needed to update prior context.

Those regressions outweighed the gains in individual response quality, so I kept the established production model.

MODEL SELECTION PRINCIPLE

The best model for a product is not necessarily the newest or most capable in isolation. It is the one that reliably preserves the behavior the product depends on.

Defining What a “Pass” Actually Means

Conversational AI created a different testing problem: two conversations can use different words, questions and turn counts while still accomplishing the same thing.

Historical transcripts therefore could not become scripts that future versions were expected to reproduce. I treated them as behavioral test cases instead.

WORDING EQUIVALENCE

Did the system ask the same questions, use the same phrasing and follow the same sequence?

BEHAVIORAL EQUIVALENCE

Did the conversation preserve the function that mattered?

A variation became a regression when it had a meaningful consequence, such as moving forward without enough context, delaying a ready evaluation, mishandling uncertainty or failing to update prior context.

This created a more durable QA standard: the language can evolve while the behaviors that define the product remain testable.

Key QA Insight: Do not test whether the conversation repeated itself. Test whether it still did its job.

The Result

The finished product is a live conversational AI experience that helps people understand what they need from work before evaluating whether an opportunity aligns with it.

What began as a job-posting scanner evolved into a broader system for reflection, work alignment and opportunity evaluation.

The production system now includes:

- guided reflection before evaluation
- adaptive behavior for different levels of user context
- a working profile that can evolve as the user changes
- opportunity evaluation grounded in individual context
- a native web interface and server-side API architecture
- production validation and failure handling
- a reusable behavioral regression suite
- a qualification process for future model changes

The full historical user-behavior suite was replayed against the live production application. Core scenarios passed, with limited conversational variation accepted where it did not change the intended function.

The result is intentionally not a system that tells someone whether to take or leave a job. It creates enough structure around reflection and evaluation to help the user make that decision with clearer context.

Explore the live Work Alignment Guide →

FINAL PRODUCT PRINCIPLE

Use AI to support better reflection, not to replace human judgment.

What I Learned

The biggest lesson from this project was that conversational AI reliability is not the same as conversational consistency.

Work Alignment Guide became more stable when I stopped expecting every conversation to follow the same path and focused instead on identifying which behaviors actually needed to remain consistent.

The project reinforced several principles I want to carry into future AI product work:

Design around behavior, not exact dialogue
Natural variation is acceptable when the system continues to perform the intended function.

Pay attention to transitions
The handoff between two individually stable behaviors can be more sensitive than either behavior on its own.

Test the complete experience
A strong individual response does not guarantee that the conversation works correctly from beginning to end.

Treat model changes as product changes
A newer model may improve response quality while introducing regressions elsewhere in an established system.

Know when to stop redesigning
Once the architecture passes its behavioral standard, observation and real user evidence can be more valuable than continued refinement.

Once the system was working reliably, I learned not to change it simply because a newer option became available.

What Comes Next

The next phase of Work Alignment Guide is less about adding features and more about learning from the product in real use.

The conversational architecture has been stabilized, migrated into a native application and validated against the behavioral scenarios developed throughout the project. Continuing to change that system without new evidence would risk solving problems that have not actually appeared.

For now, the production baseline stays intact.

Future changes should be driven by observed user behavior, recurring friction or a clear opportunity to improve the experience rather than by the assumption that a live AI product must continually become more complex.

Model upgrades will follow the same principle. A future candidate will first be tested against the established behavioral regression suite. Only a model that preserves the experience at that stage should move into validation through the native application.

Current Product → Observe Real Use → Identify Evidence → Test Change → Validate Behavior → Production

The goal for the next version is not to make Work Alignment Guide do more.

It is to make sure the next change solves something worth changing.

Next-Version Principle: Evidence before expansion.