

Pre-Implementation Diagnostic

Pre-Implementation Diagnostic

A structured intake system for monday.com Implementation Consultants

Field	Detail
Author	Charlie Fuller, Waifinder AI
Version	0.1 — Concept / Interview Draft
Status	Pre-build: designed for discussion, ready to build on confirmation
Artifact type	Credibility artifact — Implementation Consultant application, monday.com
Closest analogues	Solomon interview pipeline (SuperAssistant); DISCO triage stage (Thesis)

The Problem This Solves

Most implementation projects fail in the first two weeks, not because of platform limitations but because the consultant walked in without a clear read on the client's actual situation: their workflow complexity, their data relationships, their organizational readiness to change, and the specific friction points that will surface the moment training ends and real use begins.

The standard approach is to start with a kickoff call, ask some discovery questions, and build. The problem is that discovery and build are interleaved, which means architecture decisions happen before the picture is clear. By the time the real constraints surface — a team that has been burned by a previous tool, a reporting requirement that doesn't fit the proposed board structure, a process owner who wasn't in the room — the implementation is already underway.

This diagnostic runs before any of that. It produces two outputs that shape every decision downstream: a workflow complexity score that recommends a starting architecture pattern, and an adoption risk profile that names the specific change management landmines for this client before they detonate.

The core insight: Adoption is the product, not the platform. A consultant who diagnoses adoption risk before the first board gets built is solving the right problem. Most candidates show up with platform knowledge. This artifact shows the thinking layer underneath it.

Who It's For

Primary user: a monday.com Implementation Consultant running a new engagement. The diagnostic runs once, at the start of a project, before any configuration work begins. It takes 20-30 minutes of client input and produces a structured brief the consultant uses to scope and plan the entire implementation.

Secondary value: the outputs can be shared with the client as a project framing document, giving the engagement a clear starting contract: here is what we understand about your situation, here is the architecture we're recommending, and here are the adoption risks we are going to manage together.

System Components

1. Intake Interview

A structured, conversational intake that collects the information needed for both outputs. Delivered as a short form or a guided chat interface. Designed to feel like a smart onboarding conversation, not a questionnaire.

Input categories:

- **Current workflow:** what process is this replacing or augmenting, how is it currently managed, what tools are in use today
 - **Team structure:** size, roles, technical comfort level, prior experience with work management platforms
 - **Use case type:** project management, CRM, service desk, HR workflows, cross-functional operations, or hybrid
 - **Data relationships:** are there dependencies between items, linked records, or reporting roll-ups that require relational structure
 - **Automation appetite:** what manual steps are highest-friction, which ones are candidates for automation
 - **Integration requirements:** which external systems need to connect, what data needs to flow in or out
 - **Definition of success:** how will the client know this worked, what does good look like in 90 days
 - **Organizational context:** who owns this initiative, who are the skeptics, what happened with the last tool rollout
-

2. Workflow Complexity Score

The first structured output. Scores the engagement on four dimensions and recommends a monday.com architecture pattern as the starting point for the build.

Dimension	What it measures
Data complexity	Simple records vs. linked items, mirror columns, cross-board dependencies
Automation depth	No automation vs. basic status triggers vs. complex conditional logic and integrations
Team surface area	Single team vs. cross-functional, single board vs. portfolio view
Integration load	Standalone vs. CRM-connected, API-dependent, or multi-system data flow

Architecture patterns (recommended output):

- **Simple board:** single team, linear workflow, no cross-board dependencies
- **Relational structure:** linked items, mirror columns, cross-board roll-ups
- **Multi-board with automation:** complex workflows, conditional routing, status-driven notifications
- **Full integration play:** external system connections, bidirectional data sync, API-dependent workflows

Design decision: The score does not lock the architecture. It anchors the conversation. The consultant uses it as a starting point, not a constraint. The value is in making the complexity visible before the build begins, not in automating the architecture decision.

3. Adoption Risk Profile

The second structured output, and the one with the most direct impact on implementation success. Surfaces the specific change management risks for this client, grounded in the intake signals, not generic onboarding tips.

Risk signals the system flags:

- **Spreadsheet culture:** team has deep spreadsheet workflows and strong data ownership habits — expect resistance at the reporting and formula layer
- **Tool fatigue:** prior platform rollout failed or was abandoned — trust deficit needs to be addressed explicitly before training begins
- **Misaligned success criteria:** what the stakeholder described as success does not match the process they described — surface this before kickoff, not after go-live
- **Missing process owner:** the person driving the implementation does not have authority over the workflow being changed — adoption will stall without executive sponsorship
- **Automation overreach:** client wants to automate steps that require human judgment — scope needs to be narrowed before the build, not during it

- **Integration dependency risk:** success depends on a connection to an external system the client does not fully control — timeline and ownership need to be established upfront

What this is not: This is not a red-team exercise or a reasons-to-slow-down list. It is a proactive brief. Each flagged risk comes with a recommended conversation to have, not a recommendation to pause the project. The goal is to surface what will happen anyway — earlier, when it can be managed.

System Flow

Stage	What happens
1. Intake	Client or consultant completes the structured interview. 20-30 minutes. Can be async (form) or synchronous (guided chat).
2. Scoring	Claude processes responses and scores each complexity dimension. Pattern matching against architecture templates.
3. Risk detection	Claude maps intake signals to risk categories. Each flagged risk includes the signal that triggered it and a recommended pre-build conversation.
4. Brief generation	System produces a structured one-page brief: complexity score, recommended architecture pattern, adoption risk profile with action items.
5. Consultant review	Consultant reviews the brief, adjusts any scoring that doesn't match their read of the situation, and shares with the client as a framing document.

What Is Built vs. What Gets Built

This section is deliberately transparent. It reflects where the system stands today and where it goes next.

The engine: built and proven

The core pattern — structured intake, Claude-powered extraction, scored outputs that shape downstream architecture decisions — is the same pattern that runs in two production systems I have already shipped.

- **Solomon (SuperAssistant):** a voice interview pipeline that extracts 20+ business parameters from a 15-20 minute conversation, scores pain points on the

TRIPS framework, and deploys a personalized AI assistant configured to the executive's specific profile. The intake-to-configuration flow is production-hardened.

- **DISCO (Thesis):** a structured discovery pipeline that applies four triage criteria to validate whether a problem is worth solving, generates a GO/NO-GO decision, and produces interview guides and workshop frameworks for human facilitators before any build work begins. The pattern of diagnosing before building is the methodology.

The Pre-Implementation Diagnostic applies that same engine to a new domain: implementation consulting. The architecture is not speculative. The domain knowledge is fifteen years of implementations.

The frontier: what gets designed in this build

The adoption risk profile as a named, structured output is the part that gets designed specifically for this context. I have surfaced adoption risk in every implementation I have run — it is not new thinking. What is new is making it systematic: a defined set of risk signals, mapped to intake patterns, with recommended pre-build conversations attached to each flag.

That design work happens during the build, grounded in the intake signals the diagnostic collects. The framework for what to flag and how to frame it draws on fifteen years of change management experience. The system makes it repeatable.

The honest version of this claim: I have built the engine. I am designing the risk taxonomy for this domain. The combination is defensible in an interview because the methodology is real, the production analogues are real, and the domain knowledge behind the risk signals is real. What I am not claiming: that this system is deployed and running in a monday.com context. It is designed to be.

Build Plan

Scoped for a working prototype that can be demonstrated in a hiring manager conversation or delivered as a pre-interview artifact.

Phase 1: Intake form and Claude extraction (2-3 days)

- Build the intake form or guided chat interface covering the eight input categories
- Write the Claude system prompt for complexity scoring: dimension weighting, architecture pattern matching, output format
- Validate extraction quality against a set of test client scenarios

Phase 2: Risk detection layer (2-3 days)

- Define the risk signal taxonomy: which intake patterns map to which risk categories

- Write the risk detection prompt: signal identification, risk classification, recommended conversation framing
- Test against edge cases: client with multiple overlapping risk signals, client with very low complexity but high organizational risk

Phase 3: Brief generation and output formatting (1-2 days)

- Design the one-page brief format: complexity score visualization, architecture recommendation, risk profile with action items
- Build the output pipeline: Claude extraction to structured JSON to formatted brief
- Add consultant review layer: ability to adjust scoring before the brief is finalized

Phase 4: Demo packaging (1 day)

- Record a walkthrough of the diagnostic running on a realistic client scenario
- Prepare three to four representative outputs covering different complexity and risk profiles
- Write the one-page explainer for sharing alongside the resume

Interview Preparation

The toughest question this artifact invites:

“You built this for the application. Have you actually run something like this in a real implementation?”

Yes. The pattern is real. I ran structured pre-implementation discovery on every engagement at Waifinder and Build Solutions AI: intake conversations, explicit complexity scoring, and a deliberate step of naming the adoption risks before the build started. What I am doing here is systematizing that into a repeatable tool rather than a consultant’s judgment call. The engine that powers it, the intake-to-scored-output pipeline, is running in two production systems I built: a voice interview system for executive onboarding and a discovery pipeline for AI initiative triage. The monday.com context is new. The methodology is not.

“Why build a diagnostic instead of just showing us your implementation experience?”

Because the implementation experience is on the resume. The diagnostic shows the thinking layer underneath it. Anyone can list past projects. This shows what I believe about why implementations succeed or fail, and it shows that I have already built the system to operationalize that belief. The artifact is not a demonstration of my build skills. It is a demonstration of my implementation philosophy, made concrete.

LinkedIn Outreach Message

Applied for the Implementation Consultant role and wanted to get this in front of you directly: I built a pre-implementation diagnostic specifically for the monday.com context, a structured intake system that produces a workflow complexity score and an adoption risk profile before any configuration begins.

The underlying insight is that most implementations stall not because the platform fails but because adoption risk goes undiagnosed until it is too late to address — and I have built the engine to surface it early.

Happy to walk you through it if it would be useful.

Charlie Fuller

*Pre-Implementation Diagnostic | Charlie Fuller, Waifinder AI |
charlie@waifinder.org | Version 0.1 — Interview Draft*