

PRD — Autopilot Agent-Led Procurement Platform (Internal MVP)

Product Name

Autopilot Agent-Led Procurement Platform

Purpose

Define scope + requirements for an **internal MVP** of Autopilot's agent-led procurement experience, optimized for rapid build (hackathon style) + short stabilization.

This document optimizes for **speed and clarity**, not long-term scalability.

1. Problem Statement

Enterprise procurement for robotic cleaning and automation is slow and cognitively heavy. Buyers struggle to:

- Identify relevant options
- Quantify ROI early
- Navigate internal approvals
- Move from curiosity → commitment without friction

Autopilot's goal is to replace fragmented sales conversations + static marketplaces with a **guided, agent-led procurement experience** that:

- Educates buyers
- Structures decisions
- Generates actionable data

- Accelerates time-to-greenlight
-

2. Product Vision

Build an **industry-specific procurement platform** where users:

1. Go through **guided conversational discovery**
2. See a **dynamic marketplace / options view** update based on inputs
3. Receive an **ROI walkthrough** with basic visuals + recommendation
4. Progress to **greenlight** (sign-off path + next steps)

The experience should feel:

- Interactive, not form-based
 - Exploratory, not overwhelming
 - Assisted, not salesy
-

3. Target Users (Internal MVP)

Primary: Autopilot sales team running guided walkthroughs on live calls

Secondary: friendly design partners (post-internal MVP)

Assumption for MVP: **sales-led / guided usage** (self-serve robustness is out of scope).

4. Success Criteria

MVP is successful if:

- Sales can run Discovery → ROI → Option Selection live

- Agent reduces explanation burden on calls
- Inputs generate structured follow-up data
- Flow feels coherent (polish optional)

Not required:

- Perfect UX
 - Mobile optimization
 - Enterprise auth/security
 - Fully self-serve onboarding
-

5. Core Flow (3 Phases)

1. **Discovery** (agent-led Q&A)
 2. **ROI Walkthrough** (visualization + recommendation)
 3. **Greenlight / Checkout** (option selection + next steps + stakeholders)
-

6. Detailed Requirements

6.1 Discovery Phase

Goal: capture facility profile + priorities + constraints

UI: split-pane (Left = dynamic marketplace context, Right = agent chat)

Requirements:

- Agent asks structured questions with light gamification
- Questions adapt based on prior answers

- Inputs stored in session state
- Agent explains why it's asking a question (briefly)

Captured data examples:

- Facility type
- Scale indicators
- Priorities (cost, labor, uptime, etc.)
- Timeline/urgency

Out of scope:

- User accounts required upfront
- Heavy validation

6.2 ROI Walkthrough Phase

Goal: translate discovery into value + recommendation

Requirements:

- Generate estimated ROI/savings ranges
- Compare options (basic)
- Simple visuals (tables/charts)
- Agent narrates tradeoffs and assumptions

Notes:

- ROI logic can be hard-coded / rule-based
-

6.3 Greenlight / Checkout Phase

Goal: move from evaluation to commitment

Requirements:

- Confirm selected option
- Capture install window + stakeholders to involve (finance/IT/ops)
- Provide a “next steps” summary

Out of scope:

- payments
 - contracts
 - automated scheduling
-

7. UI / UX Structure

- Single web app
- Split view:
 - Left: dynamic visuals (options/ROI/comparison)
 - Right: agent chat

Principles:

- minimize page reloads
 - keep continuity (avoid classic multi-page funnel)
-

8. Technical Architecture (MVP)

Frontend: React or Next.js

Backend: lightweight API layer

State: session-based, simple persistence acceptable

Agent: LLM with access to session state + structured prompts

Infra: GCP or AWS; deploy frontend on Vercel

9. Explicit Non-Goals

- production-grade auth
 - mobile-first polish
 - full self-serve journey
 - advanced analytics
 - multi-tenant architecture
 - long-term data models
-

10. Execution Plan

Phase 1: optional design sprint (flows + screens)

Phase 2: 2 day in-person build hackathon

Phase 3: 1–2 weeks cleanup & stabilization

- roles: 1 jr engineer + Aaron as PM/QA/unblocker
-

11. Open Questions (resolve before build)

- Where does session state live (frontend vs backend)?
- MVP auth approach (if any)?
- Final discovery question set?
- ROI assumptions?

These must be resolved before hackathon.

12. Definition of Done (Internal MVP)

Done when:

- sales can run full flow live
 - agent drives the conversation
 - UI updates based on inputs
 - demoable to investors
-

13. Guiding Principle

Speed beats polish. Clarity beats cleverness. Working system beats perfect architecture.