

ProjectPilot AI

Agentic AI Project Management Platform

Product Requirements Document (PRD)

Version	1.0
Date	June 27, 2026
Author	Sankar Kumar Palaniappan
Status	Final

Executive Summary

ProjectPilot AI is an agentic AI-powered project management platform designed to reduce tracking overhead and accelerate team velocity. By combining intelligent Kanban boards, real-time collaboration, and AI-driven task orchestration, ProjectPilot AI enables software development teams to ship faster while maintaining full project visibility.

Primary Value Proposition: Reduce project tracking friction by 40%, improve team alignment through real-time sync, and accelerate shipping velocity by 25%.

Problem Definition

1.1 What Problem Are We Solving?

- Software development teams struggle with project tracking overhead. Current tools require manual status updates, lack real-time synchronization, and provide limited AI-driven insights into task dependencies and blockers. Teams spend 2-3 hours per week on administrative tracking instead of shipping.

1.2 Who Are We Solving For?

- **Primary Persona:** Engineering Managers (5-30 person teams) seeking autonomous task orchestration and real-time visibility
- **Secondary Persona:** Product Managers needing cross-team coordination without Slack overhead
- **Tertiary Persona:** Individual Contributors wanting intelligent task prioritization and dependency detection

1.3 Why Is This Worth Solving?

- **Market Opportunity:** \$8B project management SaaS market growing at 12% CAGR. Teams waste 15-20% of engineering capacity on process overhead. Agentic AI can reclaim this capacity through autonomous task orchestration, dependency resolution, and real-time synchronization.
- **Competitive Gap:** Existing tools (Jira, Linear, Asana) lack real-time sync and AI-driven task intelligence. First-mover advantage in autonomous project management is significant.

1.4 Why Agentic AI?

- Agentic AI enables: (1) Autonomous dependency detection across sprints, (2) Real-time task routing based on team capacity, (3) Intelligent blocker resolution via context retrieval, (4) Predictive sprint planning. This requires agent-level reasoning that static ML cannot provide.

1.5 Core Metrics

Metric	Target
Tracking Overhead Reduction	40% reduction in manual updates
Team Activation	50% of team active within 7 days
Shipping Velocity Lift	25% sprint throughput increase

Solution Definition

2.1 Core Capabilities

- **Intelligent Kanban Boards:** AI-driven card routing, automated status transitions, and dependency visualization across sprints
- **Real-Time Collaboration:** WebSocket-powered sync with conflict-free team edits and live presence indicators
- **AI Task Assistant:** Context-aware recommendations for blockers, dependencies, and optimal task sequencing
- **Multi-Team Workspaces:** Organizational hierarchy with team-level autonomy and cross-team visibility
- **Enterprise Security:** Supabase RLS, SSO integration, SOC2 compliance path, audit logging

2.2 Primary User Flow

- 1. Manager creates team workspace and imports tasks from Jira/GitHub/Linear via API
- 2. AI Agent analyzes task graph, detects blockers, and visualizes dependencies
- 3. Team members collaborate on Kanban board with real-time sync and AI suggestions
- 4. AI proactively identifies risks and recommends interventions (e.g., reduce scope, parallelize work)

Feature Specifications (V1)

3.1 Kanban Boards

- Customizable columns (Backlog, In Progress, Review, Done), drag-drop task reordering, AI-suggested card transitions, inline task editing, task cards display assignee, priority, due date, and AI risk badges

3.2 Real-Time Sync

- Supabase Realtime subscriptions with conflict-free merge (CRDT-style). Presence indicators for team members actively editing. Auditable history with 30-day rollback.

3.3 AI Assistant

- Claude LLM integration for intelligent task recommendations. Context window includes full task graph, team velocity, sprint history. Accessible via chat widget on board and task detail views. Multi-turn conversation for collaborative planning.

3.4 Security & Permissions

- Role-based access control (Admin, Manager, Contributor, Viewer) enforced via Supabase RLS. Team-level isolation with explicit cross-team sharing. Audit log for all mutations.

Non-Goals (V1)

- Time tracking, resource allocation forecasting, integrations beyond Jira/GitHub/Linear, mobile-native app, custom workflow automation, advanced reporting (data export available, BI integration deferred)

Technical Architecture

4.1 Tech Stack

Layer	Technology
Frontend	React 18, TypeScript, Tailwind CSS, Shadcn/UI
Database	Supabase (PostgreSQL + Realtime + Auth)
AI/LLM	Claude API (Anthropic) with extended context
Hosting	Vercel (Frontend), Supabase Cloud (Backend)

Success Metrics

5.1 Primary Metric

- **Board Adoption Rate:** >40% of Weekly Active Users (WAU) actively managing work via ProjectPilot within 30 days of team onboarding

5.2 Secondary Metrics

- **Team Activation:** 50% of imported team members complete onboarding and create first task within 7 days
- **Weekly Active Teams:** 15% week-over-week growth in teams with ≥3 board updates per week
- **AI Engagement:** >25% of teams use AI Assistant at least once per sprint
- **Performance:** Board load time <800ms (p95), real-time sync latency <150ms (p95)

Rollout Plan

Phase	Percentage	Duration	Criteria
Alpha	1%	2 weeks	Critical bugs resolved, 0 data loss
Beta	10%	3 weeks	>30% adoption, positive NPS
General Availability	100%	2 weeks	No P1 bugs, >40% adoption target met

Risks & Dependencies

6.1 Key Dependencies

- Claude API availability and rate limits during peak usage. Supabase Realtime stability for concurrent team collaboration. Jira/GitHub/Linear API rate limits during bulk import.

6.2 Risks & Mitigations

- **Risk:** LLM cost escalation with large task graphs. Mitigation: Implement prompt caching, batch processing, cost caps per team.
- **Risk:** Real-time sync conflicts under concurrent editing. Mitigation: CRDT-based conflict resolution, comprehensive test coverage.
- **Risk:** Adoption friction for teams habituated to Jira/Linear. Mitigation: Frictionless import, parallel operation during pilot, in-app onboarding.

Open Questions

- Should ProjectPilot support custom board workflows (beyond standard Kanban)? What is the maximum team size we need to optimize for in V1? How aggressively should AI recommendations surface (risk of alert fatigue)?

Assumptions

- Teams have ≥ 5 members for collaborative benefit. AI-driven task recommendations increase adoption $>30\%$. Supabase handles our scale (1000 concurrent users per workspace). Organizations will pay \$100-500/month for ProductPilot.