PM Strategy Copilot

Product Requirements Document · v6

Owner	TPM, Customer Use Cases
Status	Planning
Last updated	2026-02-26
Changelog	v6 — Unified experience audit (9 issues); morning workflow; agent boundaries; bidirectional data links; DayPlan history; agent team vision; additional recommendations

Owner: TPM, Customer Use Cases

Status: Planning

Last updated: 2026-02-26

Changelog: v6 — Unified experience audit applied (9 issues resolved); morning workflow formalised; agent boundaries clarified; data model bidirectional links added; DayPlan history surfaced; agent team long-term vision added; additional recommendations section added

---

1. Product Vision

PM Strategy Copilot — a focused AI system that helps a Technical Program Manager decide what matters and why.

> *The biggest value of this product is not automation. It is helping the PM make better decisions.*

The PM faces two problems every day:

Tactical overload — too many signals, not enough structure. What should I focus on? Which of these requests actually matters?

Strategic gap — no time or framework for stepping back. What patterns am I missing? Is this decision backed by data?

This system addresses both layers. Every AI output follows one rule:

Element	Meaning
What	What is happening, specifically
Why	Why it matters right now
Action	The one concrete next step

Insights without a recommended action are reports, not decisions. This system produces decisions.

---

2. Target User

A Technical Program Manager working on a single large enterprise customer engagement. Uses Microsoft Teams and Microsoft 365 daily. Comfortable with web browsers. Learning CLI over time.

Interface: Web UI (Streamlit) is the primary and default interface. CLI deferred.

---

3. Unified Experience Principles

These rules govern every design decision. When two features conflict, these principles resolve the dispute.

Rule	Meaning
One agent generates each output type	Risk signals come only from the Analyst. Day plans come only from the Planner. No two agents produce the same type of output.
One surface displays each stored record type	Insights tab shows StrategicInsights. Today tab shows DayPlans. Requests tab shows CustomerRequests. No record type appears on two tabs.
One pathway creates each data type	Chat for conversational intake. Requests tab for structured/bulk intake. No sidebar shortcuts that duplicate tab functions.
Chat is the only conversational input	No other tab has a freeform text input. Quick-action buttons on other tabs route into Chat pre-primed — they do not create separate input surfaces.
The Planner synthesises; the Analyst analyses	The Planner reads stored outputs (insights, requests) and produces a prioritised action list. The Analyst reads raw data and produces insight records. Neither does the other's job.

---

4. Feature Prioritization

Tier 1 — Build First (~4 weeks)

Defines the product. Everything else depends on these.

#	Feature	Why Tier 1
1	**Morning Workflow**	The single coherent start-of-day flow. Drives daily habit.
2	**Daily Planner Agent**	Entry point. Synthesises signals into a ranked action list.
3	**Analyst Agent**	Core strategic value. Sole generator of insight and risk signals.
4	**Request Intake & Storage**	Data layer. Agents reason about nothing without stored requests.
5	**Strategic Insight Storage + History**	Enables the feedback loop. Insights that aren't saved don't compound.

Tier 2 — Next Phase

High value. Builds on Tier 1.

#	Feature	Dependency
1	Pattern → Feature Synthesis	Needs 4–6 weeks of stored requests
2	Stakeholder Reporting (exec summary, strategic review)	Needs accumulated insights
3	Weekly Reflection Agent	Needs DayPlan history
4	Power Automate briefing automation	Nice-to-have; removes manual Copilot step

Tier 3 — Deferred

Low ROI for current context.

Feature	Reason
Roadmap management	Adds tracking overhead; not the core value
PRD / release note generation	Low relevance without multi-team scope
Customer Health dashboard	Single-customer context
CLI interface	Web UI covers all needs
Jira / Linear sync	High complexity, low v1 value
Teams bot	Requires IT/admin approval

---

5. Core Features — Tier 1

5.1 Morning Workflow

The single coherent start-of-day experience. Everything starts here.

This is not a feature — it is the primary user journey. It must feel like one fluid action, not three separate steps.

Flow:

```
PM opens app (or drops briefing file into ~/pm_agent_inbox/)
        ■
        ▼
[1] intake-agent processes the raw briefing file
        ■  Extracts and saves:
        ■  · Meetings → DayPlan.meetings_today
        ■  · Emails + Teams messages → DayPlan.context_summary
        ■  · Customer mentions → prompts PM to log as requests
```

Key rule: `intake-agent` is the sole owner of raw briefing file parsing. `planner-agent` never reads the raw file — it reads structured data that intake-agent has already extracted and stored.

5.2 Daily Planner Agent

Synthesises signals into a ranked, reasoned action list.

What it reads (inputs):

- `DayPlan.context_summary` — structured context extracted by intake-agent
- `DayPlan.meetings_today` — today's calendar
- `list_requests(priority=["P0","P1"], stale=True)` — open high-priority requests with no recent activity
- `list_insights(type="risk", confidence="high")` — recent risk signals from the Analyst

What it produces (output):

- Ranked FocusItems (3–5), each with What / Why / Source / Estimated time
- Completed `DayPlan` record saved to storage

What it does NOT do:

- Does not run its own staleness or risk analysis (that is the Analyst's job)
- Does not read the raw briefing file (that is intake-agent's job)
- Does not generate insight records (that is the Analyst's job)

Today tab — default landing page:

Focus items are checkable. When the PM marks an item done, `CustomerRequest.last_surfaced_at` and `surface_count` update. This data feeds the weekly reflection and the feedback loop.

5.3 Analyst Agent

Sole generator of strategic insight and risk signals.

Reads all stored `CustomerRequest` and `StrategicInsight` records and produces new `StrategicInsight` records. Every output follows: **What** → **Why** → **Recommended Action.**

The Analyst operates in four modes, all triggered through Chat.

Mode 1 — Trend Detection

What themes are growing in frequency?

```
TREND · High confidence · Last 30 days

What:   Audit & compliance requests up 3x in 6 weeks.
        4 requests from Stripe. 2 from their compliance team.

Why:    Stripe's Q2 audit deadline is likely driving urgency.
        No StrategicInsight records show a committed response.

Action: Add audit log to Q2 plan as P1.
        Brief engineering lead this week. Effort: M.
```

Mode 2 — Gap Detection

What is being asked for with no committed response?

In Tier 1, gap detection checks: clusters of requests with no linked `StrategicInsight` of type `"decision"` or `"trend"` indicating planned work. (Note: Gap detection does not join against `Feature` records in Tier 1 — those are Tier 2.)

```
GAP · Medium confidence · Q1 2026

What:   7 requests mention real-time notifications.
        No decision or trend insight covers this cluster.

Why:    Customers are building polling workarounds.
```

```
        Hidden support load is likely growing.

Action: Create a feature spec for notification system.
        Cross-reference request #a3f2 — may be same need.
```

Mode 3 — Risk Detection

What could escalate if not addressed?

Risk signals generated here are stored as `StrategicInsight` records. The Planner agent reads these records when building the daily focus list. This is the single source of risk signals — the Planner never independently analyses staleness.

```
RISK  ·  High confidence  ·  Now

What:   2 P0 requests open 14+ days. No update. No commitment.

Why:    P0s without visible progress risk exec escalation.
        Stripe contact last emailed 6 days ago with no reply.

Action: Send a brief status update to Stripe today.
        Flag for Q2 planning call tomorrow.
```

Mode 4 — Decision Support

Structured trade-off analysis for explicit choices.

```
DECISION  ·  Q2 Prioritisation

Options:  A) Audit log feature   B) Webhook retry (enhanced)

Signal:   Audit log: 6 requests, avg P1, growing trend
          Webhook retry: 3 requests, 1 customer (Stripe), P0

Trade-offs:
  A: Broader signal. Longer effort (L). Compliance season aligns.
  B: Single customer but production-blocking. Effort M. Fast to close.

Risks:
  A: Stripe escalation risk if webhook not addressed first.
  B: Audit log becomes urgent in ~6 weeks regardless.

Action:   Ship webhook retry first (M, unblocks Stripe).
          Queue audit log immediately after. Both fit in one quarter.
```

5.4 Request Intake & Storage

The data layer. Two explicit pathways, no others.

Pathway 1 — Conversational (Chat tab)

For single requests during the day. PM types or pastes a request in natural language. intake-agent classifies, prioritises, and saves.

> *"Log this: Stripe need webhook retry logic, blocking production rollout."*

Pathway 2 — Structured (Requests tab)

For bulk import or file-based intake.

- [+ Add request] → opens inline form (description, source)
- [Upload file ↑] → accepts CSV, PDF, Word (.docx)
- intake-agent processes uploads and presents a review list before saving

No other pathways. No sidebar [+ Request] button. No intake from Insights tab.

Per request, intake-agent automatically:

- Classifies: `feature_request | bug_report | integration | support | feedback`
- Assigns priority: `P0 | P1 | P2 | P3`
- Provides one-sentence rationale for each decision

Requests tab:

The **Surfaced** column shows `surface_count` — how many times this request has appeared in the daily plan. Zero means it has been logged but never prioritised, which is itself a signal.

5.5 Strategic Insight Storage & Reflection

The memory layer. Enables the feedback loop.

Every Analyst output is saved as a `StrategicInsight`. Over time this builds a searchable history of what patterns were spotted, what decisions were made, and what actions were recommended.

Feedback loop:

```
Analysis (Analyst detects risk)
    → Decision (Planner surfaces it as a focus item)
    → Action (PM checks it off)
    → Reflection (Past plans show it was addressed)
```

Insights tab — read-only history view:

```
■                                                           ■
■  TREND  ·  High  ·  20 Feb                                ■
■  "Audit requests up 3x in 6 weeks"                        ■
■  Action: Add audit log to Q2 as P1                       ■
■  Status: ■ Not yet acted on                               ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
```

Each insight shows whether its recommended action has been promoted into a day plan (feedback loop closing).

---

6. Data Models — Tier 1

`CustomerRequest`

```
id                        str        auto-generated 8-char ID
created_at                str        ISO timestamp
updated_at                str        ISO timestamp
deleted                   bool       soft-delete (default False)
description               str
raw_input                 str        original text before processing
source                    Literal    "chat" | "csv" | "pdf" | "docx" | "copilot_briefing"
source_ref                str        filename or "typed"
classification            Literal    "feature_request" | "bug_report" | "integration"
                                     | "support" | "feedback"
classification_rationale str        one-sentence AI explanation
priority                  Literal    "P0" | "P1" | "P2" | "P3"
priority_rationale        str        one-sentence AI explanation
status                    Literal    "new" | "triaged" | "in_review" | "linked" | "closed"
                                     ("linked" = associated with a Feature, Tier 2)
tags                      list[str]
last_surfaced_at          str | None ISO timestamp — last time this appeared in a FocusItem
surface_count             int        how many DayPlans this request has appeared in
linked_insight_ids        list[str]  StrategicInsights that reference this request
edit_history              list[dict] [{timestamp, field, old_value, new_value}]
```

`DayPlan`

```
id                        str
date                      str        "YYYY-MM-DD"
generated_at              str        ISO timestamp
briefing_source           str        filename or "pasted"
focus_items               list[FocusItem]
context_summary           str        structured summary extracted by intake-agent
meetings_today            list[dict] [{title, start_time, duration_min}]
```

`FocusItem` (embedded in DayPlan)

```
rank                      int        1–5
title                     str
what                      str        the specific action
why                       str        why it matters today
source_type               Literal    "customer_request" | "calendar" | "email"
                                     | "teams" | "insight" | "backlog"
source_ref                str        non-request sources (meeting title, email subject,
                                     insight ID). Not used for request links.
linked_request_ids        list[str]  CustomerRequest IDs driving this item
```

```
  estimated_minutes       int
  done                    bool       marked complete by PM (default False)
```

`StrategicInsight`

```
  id                      str
  created_at              str
  insight_type            Literal    "trend" | "gap" | "risk" | "decision"
  title                   str
  what                    str
  why                     str
  recommended_action      str
  confidence              Literal    "high" | "medium" | "low"
  period                  str        "last-30-days" | "2026-Q1" etc.
  linked_request_ids      list[str]
  in_day_plan             bool       True if promoted into any FocusItem (feedback loop)
```

Confidence definitions (for consistent AI output)

Level	Meaning
High	3+ requests with same theme, or a P0 with explicit urgency signal
Medium	2 requests with related theme, or inferred urgency without explicit signal
Low	Single request, or pattern is speculative

Pre-defined schemas (not activated in Tier 1)

- `Feature` — Tier 2 (Pattern → Feature Synthesis)
- `RoadmapItem` — Tier 3
- `Report` — Tier 2

---

7. Agent Design — Tier 1

Orchestrator

- Persistent multi-turn session — preserves context across messages
- Routes user intent to the right agent based on trigger patterns below
- Handles simple read queries (list requests, show insights) directly without delegating
- Labels every response with the agent that produced it: `[Planner]`, `[Analyst]`, `[Intake]`

Routing rules:

Intent signal	Routes to	Example
Briefing file detected / "plan my day"	intake-agent → then planner-agent	File drop, "start my day"

Intent signal	Routes to	Example
Logging a request, file upload	intake-agent	"Log this", CSV drop, "Stripe said..."
Pattern / trend / gap / risk query	analyst-agent	"What's trending?", "What's at risk?"
Decision / trade-off query	analyst-agent (decision mode)	"Help me decide between X and Y"
Ambiguous intent	Orchestrator asks: "Do you want to log a request or analyse your backlog?"	"What about SSO?"

Three agents in v1

Agent	Job	Reads	Writes
`intake-agent`	Parses raw inputs. Classifies requests. Extracts briefing structure.	Raw files (CSV, PDF, docx, briefing md)	CustomerRequest, DayPlan.context_summary, DayPlan.meetings_today
`planner-agent`	Synthesises stored signals into a ranked daily action list.	DayPlan (context + meetings), CustomerRequests (P0/P1 stale), StrategicInsights (risk, high confidence)	DayPlan (focus_items), CustomerRequest.last_surfaced_at, CustomerRequest.surface_count
`analyst-agent`	Detects patterns, gaps, and risks. Supports decisions.	CustomerRequests (all), StrategicInsights (all)	StrategicInsight

Critical boundary: `planner-agent` never reads raw files. `analyst-agent` never reads raw files. Only `intake-agent` reads raw files.

---

8. Microsoft Copilot Integration

Why Copilot, not Graph API

	Graph API	Copilot file drop
Setup	Azure app, OAuth, possible IT approval	None — you already have Copilot

	Graph API	Copilot file drop
Data access	Limited by API scopes	Full — transcripts, all channels, SharePoint
Data quality	Raw JSON to parse	Pre-summarised by AI
Maintenance	Token expiry, rate limits	None

The Copilot Daily Briefing Prompt

Run this in Microsoft Copilot each morning. Paste the output into Chat, or save as `briefing_YYYY-MM-DD.md` in `~/pm_agent_inbox/`.

```
Generate my daily PM briefing for today. Use exactly this markdown
structure — my PM tool parses it automatically:

# Daily PM Briefing — [TODAY'S DATE]

## Today's Meetings
For each meeting on my calendar today:
- **[HH:MM] [Title]** (duration) — Attendees: [names]
  Note any customer mentions. Key agenda if visible.

## Emails — Last 24 Hours
Emails needing my attention (skip newsletters, auto-notifications):
- **From:** [sender] | **Subject:** [subject] | **Received:** [HH:MM]
  Summary: [1 sentence]. Action needed: [Yes / No / FYI]

## Teams Messages — Last 24 Hours
Direct messages and @mentions not yet replied to:
- **From:** [sender] | **In:** [chat or channel] | **Time:** [HH:MM]
  Summary: [1 sentence]. Action needed: [Yes / No]

## Customer Mentions
Specific customer company mentioned anywhere today:
- **[Company]** — Where: [meeting/email/Teams].
  Context: [1 sentence]. Sentiment: [Positive/Neutral/Concern/Urgent]

## Action Items & Commitments
Explicit asks or commitments from the last 24 hours:
- [What] — From/for: [who] — Due: [date / ASAP / unclear]

## Open Flags
Anything urgent that doesn't fit above:
- [Item]

Focus on my role as a TPM managing a customer engagement. Concise.
```

---

9. Interface Design

Four tabs. One job each.

Tab	Job	Input?
Today	Show today's plan. View past plans. Check off items.	Checkboxes only
Chat	Talk to any agent. The only conversational input surface.	Yes — freeform text
Insights	Browse saved StrategicInsights. Quick-action buttons pre-prime Chat.	Buttons only (no text box)
Requests	View, add, and upload customer requests.	Form + file upload

Layout

Sidebar shows status only — no action buttons. Action buttons belong on the relevant tab. Sidebar [+ Request] button is removed.

---

10. Long-Term Vision — The Agent Team

v1 is the foundation. The PM Strategy Copilot is designed to grow into a full personal AI team — each agent a specialist, all coordinated by the PM Agent supervisor.

Agent	Role	When to build
Chief of Staff	Planner-agent evolved: day planning, meeting prep, commitment tracking	v1 (current)
Strategist	Analyst-agent evolved: frameworks, problem structuring, Socratic reasoning	v1 (current)

Agent	Role	When to build
Challenger	Red-teams decisions; argues the opposing view; stress-tests plans	Tier 2
Writer	Drafts emails, Teams messages, exec briefs; edits for tone and clarity	Tier 2
Researcher	Deep dives: competitive landscape, industry trends, customer context	Tier 2
Coach	PM skill development; feedback on decisions; career strategy	Tier 3
Historian	Institutional memory; past decisions and outcomes; precedent retrieval	Tier 3
Mirror	Reflects behavioural patterns back to the PM; surfaces blind spots	Tier 3

The PM Agent supervisor — the Orchestrator — remains the single coordination layer across all agents at every phase. No agent talks to the PM directly without the Orchestrator's involvement.

---

11. Additional Recommendations

Items not in the current build scope but worth considering before or during implementation.

R1 — Onboarding flow (High priority)

Day 1 without guidance is confusing. A first-run setup wizard should walk the PM through: (1) saving and running the Copilot prompt, (2) creating `~/pm_agent_inbox/`, (3) importing an initial batch of requests via CSV. Without seed data, the Analyst has nothing to reason about and the experience is thin for the first week.

R2 — Context persistence across sessions (High priority)

The PRD defines a multi-turn session, but sessions end when the app closes. Tomorrow's planner needs to know what was planned today and whether items were completed. The conversation history and `DayPlan.done` states must persist between sessions, not just within one. Design this in Week 1 — retrofitting session persistence is painful.

R3 — Weekly reflection (Tier 2, low effort)

Once DayPlan history exists, a weekly reflection prompt becomes trivial to add. Every Friday: "Here's what your week looked like — 5 plans, 18 focus items, 12 checked done. Recurring item: SSO integration appeared every day. Still open." This closes the feedback loop at a weekly grain, not just daily.

R4 — Agent response labelling (Implement in v1)

Every response from a sub-agent should be prefixed with its name: `[Planner]`, `[Analyst]`, `[Intake]`. This is a single-line implementation but has outsized UX value — it teaches the PM what each agent does through daily use, and it makes multi-agent collaboration visible rather than opaque.

R5 — Privacy boundary declaration (Document before launch)

The briefing file contains meeting attendees, email content, and Teams messages. All data must stay local — no cloud sync, no telemetry, no data sent outside the machine except to the Claude API for inference. This should be explicitly stated in the Settings tab and the README. In an enterprise context, a PM needs to be able to answer "where does my data go?" before trusting the tool.

R6 — Graceful first week (Design consideration)

The Analyst's quality degrades sharply with fewer than ~10 stored requests. Trend detection on 3 requests is noise, not signal. Consider two mitigations: (1) the onboarding CSV import seeds at least 10–15 historical requests; (2) the Analyst shows a "low data" warning when fewer than 10 requests exist, so the PM knows to interpret outputs cautiously.

R7 — The Challenger agent as early Tier 2 priority

Of the extended agent team, the Challenger has the highest near-term leverage and lowest build cost. It reuses the Analyst's data access and adds a single system prompt change — "argue the opposing view." Once the Analyst agent is working well (Week 2), the Challenger can be added as Mode 5 of the same agent with minimal incremental effort.

---

12. Key Dependencies

```
anthropic           # Claude API
pydantic>=2.0       # Data models and validation
streamlit>=1.35     # Web UI
rich>=13.0          # CLI formatting (Tier 3 ready)
pypdf>=4.0          # PDF text extraction
python-docx>=1.0    # Word document (.docx) extraction
watchdog>=4.0       # Watch inbox folder for new briefing files
anyio>=4.0          # Async runtime
```

---

13. Build Sequence — 4 Weeks

Week 1 — Foundation + Morning Workflow

- All Tier 1 schemas (CustomerRequest, DayPlan, FocusItem, StrategicInsight) with new fields
- StorageManager (atomic writes, soft-delete, bidirectional link updates)
- `intake-agent` — briefing file parsing, CSV/PDF/Word ingestion
- `planner-agent` — reads structured context, produces FocusItems, saves DayPlan
- Streamlit: **Today tab** (with checkboxes + History) + **Requests tab** (form + upload)

- *Deliverable: Paste Copilot briefing → morning workflow runs → Today tab shows plan*

Week 2 — Analyst Agent

- StrategicInsight storage + Insights tab (read-only, quick-action buttons)
- `analyst-agent` — Trend, Gap, Risk, Decision modes
- Orchestrator routing rules + agent response labelling
- Streamlit: **Chat tab** + **Insights tab**
- *Deliverable: "What's at risk?" → structured RISK insight saved and displayed*

Week 3 — Polish + Feedback Loop

- FocusItem checkboxes update `CustomerRequest.last_surfaced_at` + `surface_count`
- Insight `in_day_plan` flag set when insight is promoted by Planner
- Surfaced column in Requests tab
- Status indicator on Insights tab (● In plan / ■ Not acted on)
- *Deliverable: Full feedback loop visible — plan → check off → insight status updates*

Week 4 — Refinement

- Session persistence (conversation history + DayPlan state across restarts)
- Onboarding flow (first-run setup wizard)
- Output quality tuning across all three agents
- Privacy statement in UI
- *Deliverable: Complete, polished Tier 1 system ready for daily use*

---

14. Cross-Check

Does Tier 1 answer the PM's core questions?

Question	Answered	By
What should I focus on today?	[✓]	Planner → Today tab
Which request is most urgent?	[✓]	P0/P1 classification + surface_count
What patterns am I missing?	[✓]	Analyst trend + gap detection
What's at risk of escalating?	[✓]	Analyst risk detection → promoted to focus list
Is this decision backed by data?	[✓]	Analyst decision mode + signal counts
Did I act on last week's insights?	[✓]	Insight in_day_plan flag + DayPlan history

Key success metric

The system succeeds if the PM opens it every morning and uses it to structure their day. Daily usage is the only metric that matters in the first 4 weeks.