AI for Product Managers - Training Workshop

Workshop Overview

Goal: Help PMs develop deep AI understanding so they can evaluate how modern AI tools apply to Product Mangement and to building products.

Path: Build intuition through personal AI productivity and use case exploration **first**, then apply that understanding to product decisions.

Format: Single 60-90 minute session showcasing the full spectrum, with optional follow-up sessions for deeper dives.

What They'll Leave With:

- Hands-on experience with AI tools (Claude.ai / Claude Code, ChatGPT, voice input, M365 Copilot)
- A mental model for "where can AI help in my workflow?"
- Framework for evaluating AI features in your own products
- Reference playbook with 12 PM workflows they can try, build on and extend themselves

---

Workshop Design Philosophy

This workshop follows an intentional and repeatable format:

The 90-Minute Workshop Flow

```
┌─────────────────────────────────────────────────────────────┐
│ 1. BOILERPLATE INTRO (15–20 min)                            │
├─────────────────────────────────────────────────────────────┤
│ • Frame the AI landscape — what's changed, what's possible  │
│ • Introduce key mental models (chatbot → agent shift)       │
│ • Teach prompting basics (CRAFT framework)                  │
│ • Set expectations: "build intuition, not expertise"        │
│                                                             │
└─────────────────────────────────────────────────────────────┘
                          │
                          ▼
┌─────────────────────────────────────────────────────────────┐
│ 2. PM USER STORIES + AI TRANSFORMATION (reference material) │
├─────────────────────────────────────────────────────────────┤
│ Document specific PM workflows as user stories:             │
│ • "As a PM, I need to [task] so I can [outcome]"            │
│ • Show before/after: traditional approach vs. AI-augmented  │
│ • Include workflow sketches with clear steps                │
│ • Estimate time savings (makes value concrete)              │
```

Aim for 10–15 user stories covering the PM skill spectrum:
Discovery, Definition, Delivery, Communication, Analysis

These become the reference playbook attendees take home.

▼

3. LIVE DEMOS — SUBSET OF STORIES (30–40 min)

Pick 3–4 workflows to demo live. Selection criteria:

- Progression of complexity:
  Basic prompting → Voice input → Tool integration → Agent
- Mix of "wow factor" and "I could do this today":
  - At least one that's immediately actionable
  - At least one that shows the ceiling of what's possible
- Relatable to everyone in the room:
  - Interview synthesis, meeting notes = universal PM pain
  - Prototyping, artifact updates = visual and concrete

Prepare speaker scripts with timing for each demo.

▼

4. CAPSTONE — ORG-SPECIFIC COMPOUND WORKFLOW (10 min)

End with a demo that:
- Solves a REAL problem the organization has struggled with
- Combines multiple AI capabilities shown earlier
- Shows the "ceiling" — what's possible with investment
- Inspires the question: "What could WE automate?"

Examples by org type:
- Retail/POS company → Standards conformance automation
- Healthcare → Compliance documentation generation
- Fintech → Regulatory change impact analysis
- SaaS → Customer feedback → roadmap prioritization

Key message: "A PM with enough curiosity could build this."

▼

5. CLOSING + SCHEDULE DEEPER DIVES (15 min)

- Summarize the toolkit (what to use for what)
- Provide evaluation framework for AI in products
- Share the playbook document for self-study

```
| • Q&A focused on "what would you automate?"                  |
|                                                              |
| Offer follow-up sessions:                                    |
| • "Vibe Coding Deep Dive" — hands-on prototyping             |
| • "AI for Your Products" — evaluation frameworks             |
| • "Building Your First Workflow" — Claude Code basics        |
| • Office hours for individual workflow design                |
|                                                              |
| The 90-min session is the appetizer. Deeper dives are        |
| where real skill-building happens.                           |
└──────────────────────────────────────────────────────────────┘
```

Why This Structure Works

1. **Boilerplate intro grounds everyone** — Even if AI fluency varies, everyone starts with the same mental models.
2. **User stories make it concrete** — Abstract AI capabilities become relatable when framed as "here's a task you already do."
3. **Live demos build belief** — Seeing is believing. Demos convert skeptics faster than slides.
4. **Org-specific capstone creates ownership** — When the example is *their* problem, they see themselves using this.
5. **Follow-up sessions enable depth** — 90 minutes creates awareness; deeper dives create capability.

PART 1: PREAMBLE SLIDES (15-20 minutes)

Slide 1: Title + Framing

SLIDE TITLE: AI for Product Managers: From Personal Productivity to Product Strategy

CONTENT:

> "You can't design AI experiences you don't understand intuitively."

Today's Journey:

1. **Build intuition** — Use AI to supercharge your own PM work
2. **Understand capabilities** — What AI is good at, where it struggles
3. **Apply to products** — How to evaluate AI opportunities for iQmetrix

SPEAKER NOTES:
The goal isn't to make you AI experts. It's to give you enough hands-on experience that when someone proposes "let's add AI to feature X," you can ask the right questions and make smart decisions.

Slide 2: The AI Landscape Today

SLIDE TITLE: The Tools Landscape (Late 2025)

CONTENT:

Chat Interfaces (What most people know):

- ChatGPT, Claude.ai, Gemini — Conversational AI for Q&A, drafting, analysis
- M365 Copilot — AI embedded in Word, Excel, Outlook, Teams (you have this!)

Code writing agents that have been repurposed for waayyyyy more:

- GitHub Copilot — Code completion and generation
- Cursor, Windsurf — AI-native code editors
- Claude Code — Rapidly become an all-purpose agent for many tasks beyond coding

Prototyping & Design:

- Lovable, Bolt, etc. — Generate working UI from descriptions
- Figma AI / Make — Design-to-code workflows

Voice & Multimodal:

- Whisper-based tools — Voice-to-text for hands-free AI interaction
- Vision capabilities — Analyze screenshots, diagrams, documents

The Trend: Moving from "chat with AI" → "AI as autonomous agent that takes actions"

SPEAKER NOTES:
You don't need to know all of these. The key insight: AI is moving from "assistant you talk to" toward "agent that does work." We'll show you what that means practically.

Slide 3: What Are AI Agents?

SLIDE TITLE: From Chatbots to Agents: The Key Shift

CONTENT:

Traditional AI (Chatbot):

- You ask a question → AI responds
- You copy the answer → paste into your tool
- Repeat for each step
- AI has no memory between sessions

AI Agent:

- You describe a goal → AI breaks it into steps
- AI takes actions (searches, reads files, calls APIs, writes documents)

- AI maintains context across the entire task
- You supervise and approve, agent executes

Simple Example:
Chatbot Approach	Agent Approach
"Write me a PRD" → copy to Confluence	"Create a PRD in Confluence for feature X" → done
"Now create Jira tickets" → copy each one	Agent reads PRD, creates tickets, links them
15 minutes of copy-paste	2 minutes of supervision

SPEAKER NOTES:
This is the mental model shift. Think of it like going from "AI as smart Google" to "AI as junior team member who can follow instructions and use tools."

Slide 4: What Agents Are Good At (GDPVal Evidence)

SLIDE TITLE: Building Intuition: What Agents Can Actually Do

CONTENT:

OpenAI's GDPVal Study: Agents performing as well as human evaluators across retail trade tasks.

Example: Retail Demand Forecasting

Metric	Human	Agent
Time to complete	2.5 hours	8 minutes
Accuracy (MAPE)	15%	14%
Factors analyzed	4-6	12+ simultaneously
Explainability	Notes	Full reasoning provided

Example: Root Cause Analysis

Scenario: Sudden 40% spike in returns for electronics category

Traditional Approach	Agent Approach
Data analyst pulls reports (2 days)	Scanned 15 data sources simultaneously
Cross-functional meeting (3 days to schedule)	Identified correlation in minutes
Hypothesis testing (1 week)	Generated 3 ranked hypotheses with evidence
Total: 10-14 days	**Total: 12 minutes**

SPEAKER NOTES:
These aren't lab experiments — these are production systems. The pattern: agents excel at pattern recognition, hypothesis generation, and multi-source synthesis. Humans excel at judgment on edge cases and strategic decisions.

Slide 5: The Agent Capability Spectrum

SLIDE TITLE: Where Agents Excel vs. Where Humans Are Essential

CONTENT:

Agents Excel At:

- Processing large volumes of information quickly
- Finding patterns across multiple data sources
- Generating first drafts and variations
- Repetitive tasks with clear rules
- Maintaining consistency across documents
- 24/7 availability, no context-switching cost

Humans Are Essential For:

- Judgment on ambiguous situations
- Stakeholder relationships and politics
- Creative leaps and novel ideas
- Ethical considerations and values
- Final accountability for decisions
- Understanding unstated context

The Sweet Spot:

> Agent does the legwork → Human makes the call

SPEAKER NOTES:
The goal isn't replacement. It's leverage. A PM who used to spend 4 hours on competitive research can now spend 30 minutes reviewing and refining agent-generated analysis.

Slide 6: The Prompting Mental Model

SLIDE TITLE: How to Talk to AI: The CRAFT Framework

CONTENT:

Context — Background information the AI needs
Role — Who should the AI be? (analyst, writer, critic)
Ask — The specific request
Format — How you want the output structured
Tone — Voice and style guidelines

Example:

> **Context:** I'm a PM at a retail POS company. We're evaluating whether to add AI-powered inventory recommendations.
>
> **Role:** Act as a senior product strategist who has shipped AI features at enterprise SaaS companies.
>
> **Ask:** Identify the top 3 risks of this feature and how to mitigate each.
>
> **Format:** Bullet points with risk, likelihood, impact, and mitigation for each.
>
> **Tone:** Direct and practical, no fluff.

Pro Tips:

- Be specific about what you DON'T want
- Include examples of good output when possible
- Iterate — first response is rarely final

SPEAKER NOTES:
You don't need to use this framework explicitly every time. But when you're not getting good results, walk through CRAFT to diagnose what's missing.

Slide 7: Voice Input — Lowering the Barrier

SLIDE TITLE: Talk, Don't Type: Voice-First AI Interaction

CONTENT:

Why Voice Matters:

- Faster than typing (3x for most people)
- More natural for brainstorming and ideation
- Captures nuance and tone you'd edit out when typing
- Enables AI use while walking, commuting, thinking

Free Tools:

- **MacWhisper** (Mac) — Local transcription, privacy-friendly
- **Whisper.cpp browser tools** — Web-based, no install
- **Claude mobile app** — Built-in voice input

- **M365 Copilot voice** — In Teams and mobile apps

Workflow Pattern:

1. Voice-capture your raw thoughts (2 min ramble)
2. AI transcribes and structures
3. You refine the structured version
4. Result: 10 minutes vs. 45 minutes of writing

SPEAKER NOTES:
This is a game-changer for ideation. You can literally think out loud and get structured output. We'll demo this with interview synthesis.

Slide 8: Today's Journey — 12 PM Workflows

SLIDE TITLE: The PM Workflow Playbook

CONTENT:

Discovery Phase:

1. Competitor Analysis
2. Interview Synthesis ⭐ *DEMO*
3. Rapid Domain Learning

Definition Phase:
4. Feature Ideation
5. Vibe Coding Prototypes ⭐ *DEMO*
6. PRD → Jira Tickets (MCP)

Delivery Phase:
7. Confluence/Aha Updates (MCP) ⭐ *DEMO*
8. Meeting Synthesis

AI Product Thinking:
9. AI Feature Evaluation
10. AI Debugging/Evals

Capstone: iQmetrix specific use case that blends all these things together
11. TMForum Standards Automation ⭐ *DEMO*

Bonus:
12. Custom Learning Path Generation

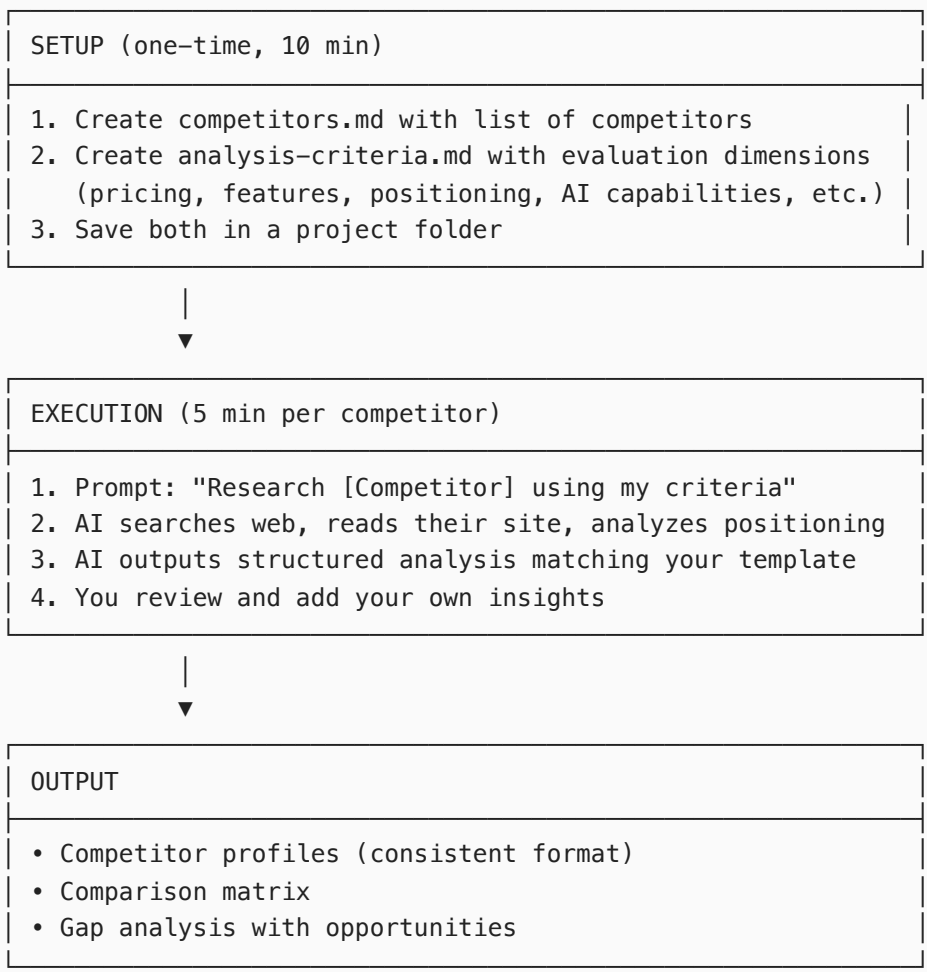
SPEAKER NOTES:
We'll demo 4 of these live. The full playbook has workflow sketches for all 12 that you can try yourself after the session.

PART 2: THE 12 PM WORKFLOWS

Workflow 1: Competitor Research

User Story: *"As a PM, I need to understand what competitors are doing in AI-powered retail analytics so I can identify gaps and opportunities."*

Workflow Sketch:



Tools: Claude.ai or ChatGPT with web search enabled

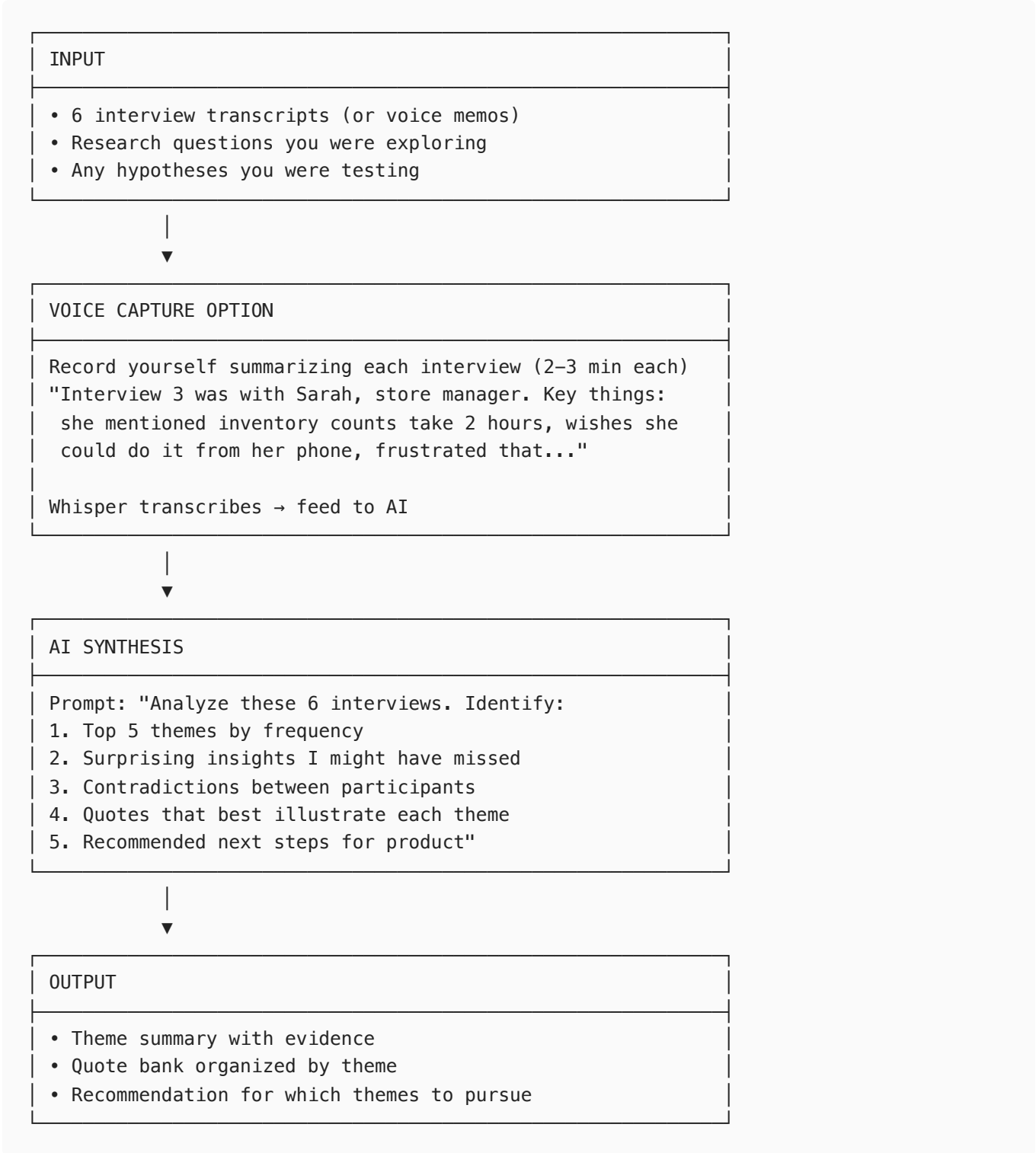
Time Savings: 4-6 hours → 45 minutes

Key Insight: The reusable setup (criteria + competitor list) means subsequent analyses take minutes, not hours.

Workflow 2: User Interview Synthesis ⭐ RECOMMENDED DEMO

User Story: *"As a PM, I need to synthesize 6 customer interview transcripts into actionable themes so I can identify patterns."*

Workflow Sketch:



Tools: MacWhisper or voice input + Claude.ai

Time Savings: 3-4 hours → 30 minutes

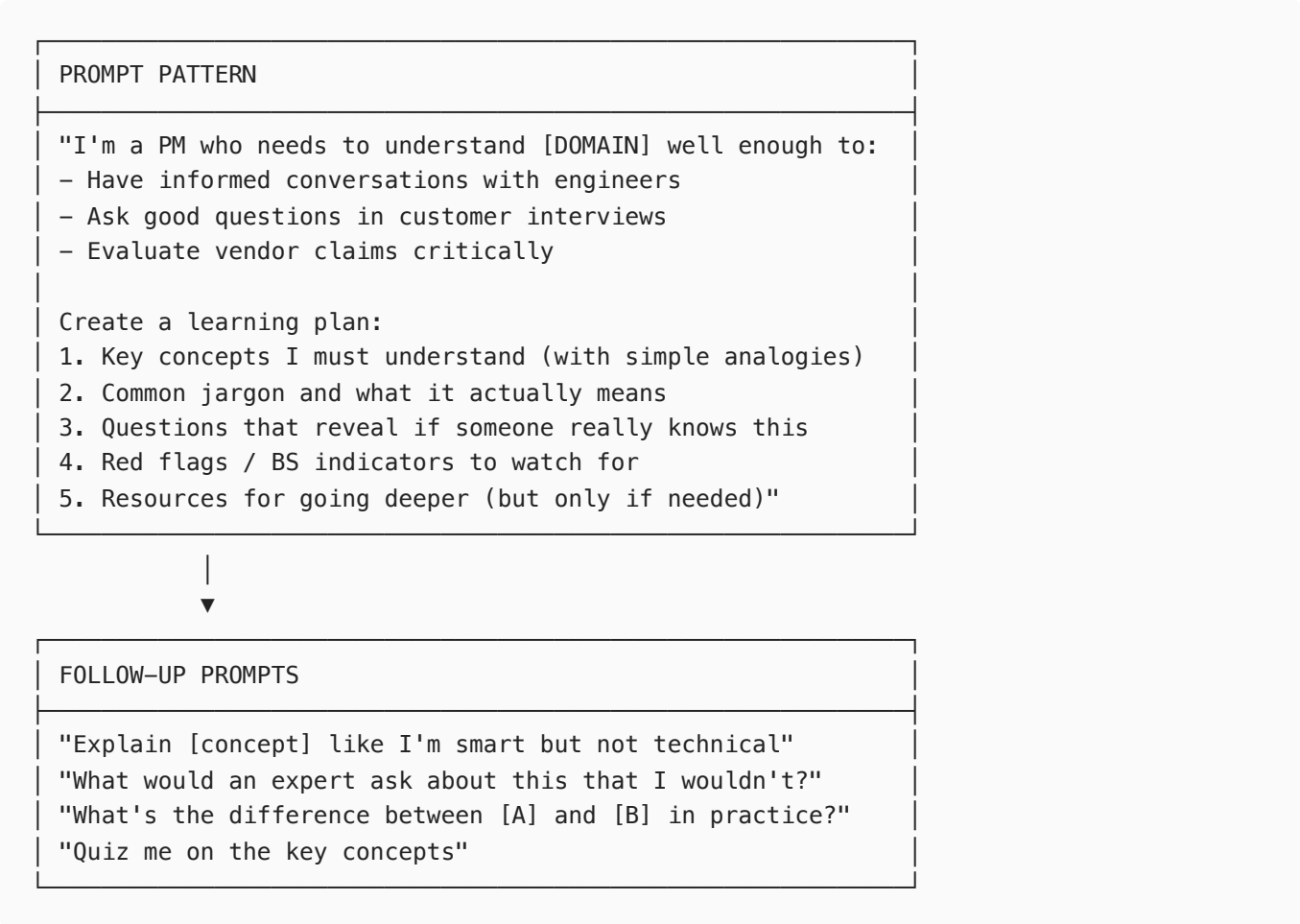
Why Demo This:

- Shows voice input (lowering the barrier)
- Relatable to every PM
- Clear before/after
- Quick to demonstrate

Workflow 3: Rapid Domain Learning

User Story: *"As a PM, I need to quickly get up to speed on a new domain (e.g., inventory optimization algorithms) so I can have informed conversations with engineers."*

Workflow Sketch:



Tools: Claude.ai or ChatGPT

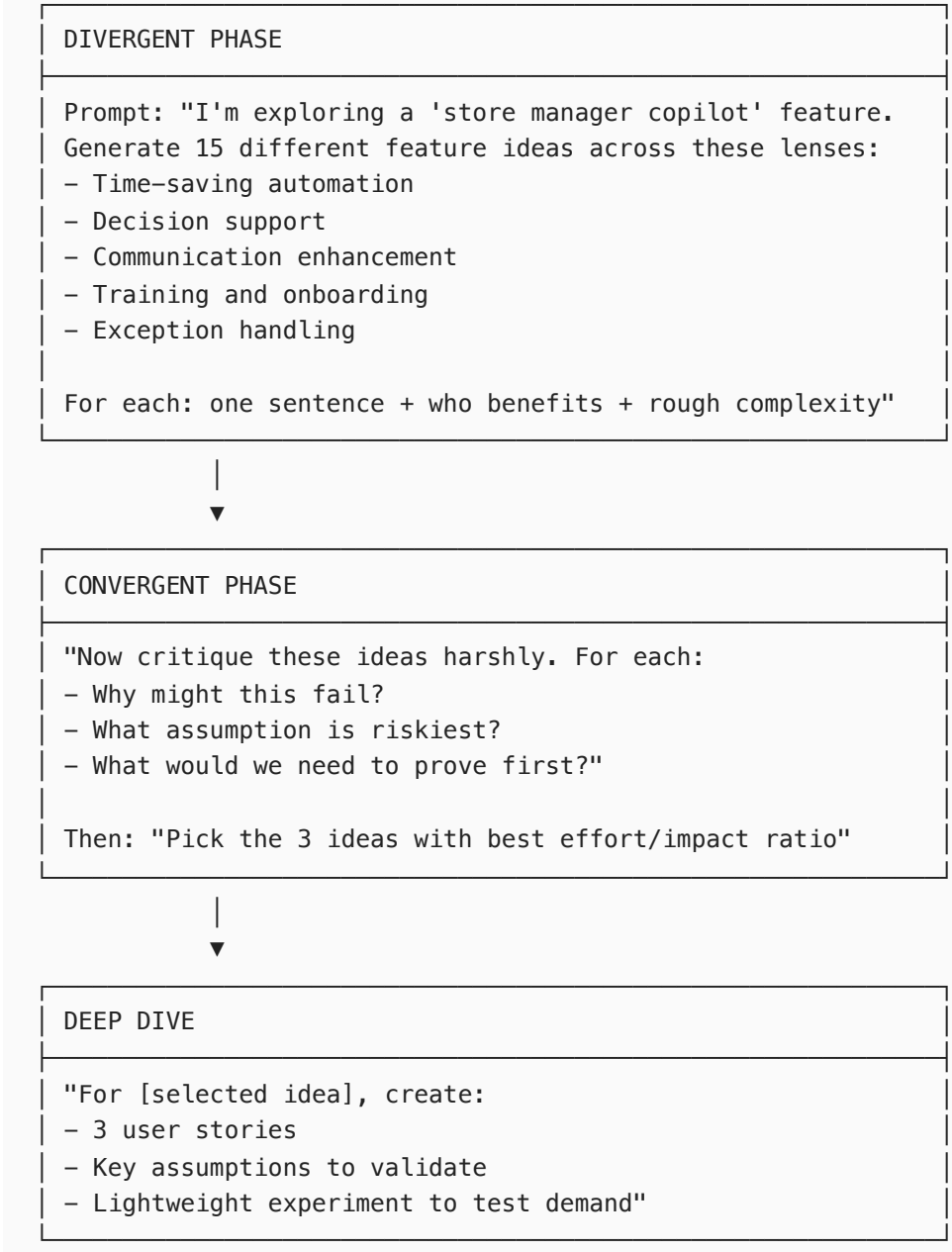
Time Savings: Days of research → 1-2 hours of conversation

Key Insight: The "quiz me" follow-up is powerful — it reveals gaps in your understanding before you're in a meeting.

Workflow 4: Feature Ideation & Brainstorming

User Story: *"As a PM, I need to brainstorm feature ideas and explore different directions for a 'store manager copilot'."*

Workflow Sketch:



Tools: Claude.ai (strong reasoning helps with critique phase)

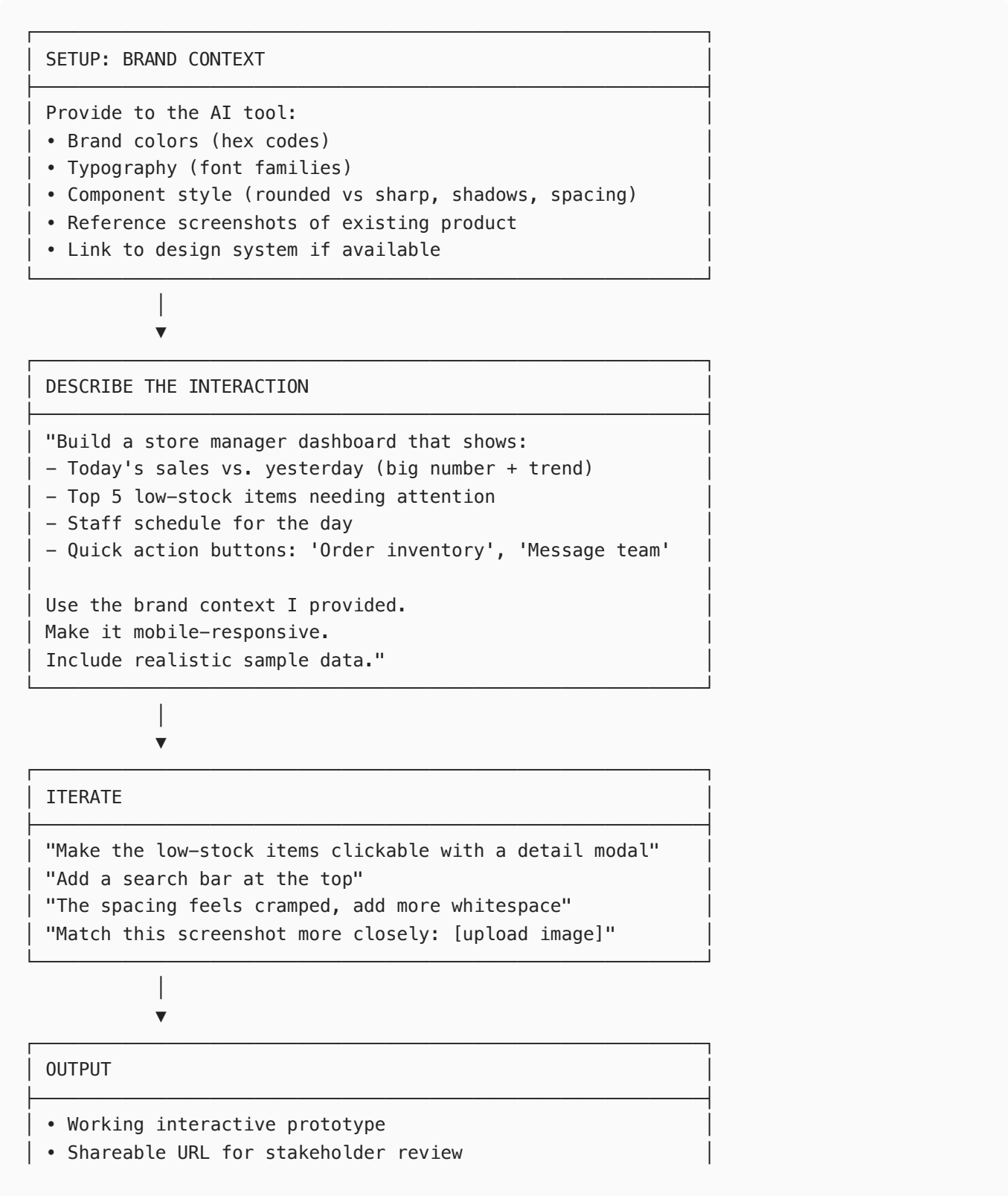
Time Savings: Half-day ideation session → 45 minutes

Key Insight: The AI is useful for both generating AND critiquing. Ask it to argue against its own ideas.

Workflow 5: Vibe Coding — Interactive Prototypes ⭐ RECOMMENDED DEMO

User Story: *"As a PM, I need to show my developers and designers what I have in mind with a real, interactive prototype that matches our brand."*

Workflow Sketch:



```
│ • Code export for engineering reference                    │
└────────────────────────────────────────────────────────────┘
```

Tools:

- **v0.dev** (Vercel) — Best for React/Next.js components, free tier
- **Lovable** — Full app generation, good for multi-page flows
- **Bolt.new** — Quick prototypes, runs in browser
- **Figma Make** — Design-to-code from Figma files
- **Claude Code + MCP** — Most powerful but requires terminal comfort

Time Savings: 2-3 design/dev cycles → 30 minutes to working prototype

Why Demo This:

- Visual "wow" factor
- Shows AI creating real, working UI
- Brand consistency angle is sophisticated
- Something most PMs don't know is possible

Demo Script: See detailed playbook below.

Workflow 6: PRD → Jira Tickets (MCP Integration)

User Story: *"As a PM, I need to write a clear PRD and automatically create the Jira tickets from it."*

Workflow Sketch:

```
┌────────────────────────────────────────────────────────────┐
│ STEP 1: DRAFT PRD                                          │
├────────────────────────────────────────────────────────────┤
│ Use standard prompting to draft PRD:                       │
│ "Help me write a PRD for [feature]. Include:               │
│ - Problem statement                                        │
│ - User stories                                             │
│ - Acceptance criteria                                      │
│ - Out of scope                                             │
│ - Success metrics"                                         │
│                                                            │
│ Iterate until satisfied.                                   │
└────────────────────────────────────────────────────────────┘
                          │
                          ▼
┌────────────────────────────────────────────────────────────┐
│ STEP 2: EXTRACT TICKETS                                    │
├────────────────────────────────────────────────────────────┤
│ "Based on this PRD, create Jira tickets:                   │
```

```
│ - One epic for the feature                                    │
│ - Stories broken down by user-facing capability               │
│ - Each story has acceptance criteria                          │
│ - Estimate complexity (S/M/L)                                 │
│ - Identify dependencies between stories"                      │
└──────────────────────────────────────────────────────────────┘
                          │
                          ▼
┌──────────────────────────────────────────────────────────────┐
│ STEP 3: CREATE IN JIRA (with MCP)                             │
├──────────────────────────────────────────────────────────────┤
│ With Jira MCP server connected:                               │
│ "Create these tickets in project [PROJECT_KEY]"               │
│                                                               │
│ AI creates epic, creates stories, links them,                 │
│ sets fields, returns links to created tickets.                │
└──────────────────────────────────────────────────────────────┘
```

Tools:

- Claude.ai for PRD drafting
- Claude Code + Atlassian MCP for Jira creation (or manual for now)

Time Savings: 2-3 hours → 30 minutes

Key Insight: Even without MCP, the "extract tickets from PRD" step saves massive time. MCP just eliminates copy-paste.

Workflow 7: Confluence/Aha Artifact Updates (MCP) ⭐ RECOMMENDED DEMO

User Story: *"As a PM, I need to update our Confluence roadmap and status documents based on what shipped this sprint."*

Workflow Sketch:

```
┌──────────────────────────────────────────────────────────────┐
│ CONTEXT GATHERING                                             │
├──────────────────────────────────────────────────────────────┤
│ AI reads from connected tools:                                │
│ • Jira: Completed tickets this sprint                         │
│ • Git: Merged PRs and their descriptions                      │
│ • Slack: Release announcements (optional)                     │
└──────────────────────────────────────────────────────────────┘
                          │
                          ▼
┌──────────────────────────────────────────────────────────────┐
│ SYNTHESIS                                                     │
```

```
┌─────────────────────────────────────────────────────────┐
│ "Based on what shipped, draft updates for:              │
│ 1. Release notes (customer-facing language)             │
│ 2. Roadmap status (move items, update dates)            │
│ 3. Sprint summary (internal team view)"                 │
└─────────────────────────────────────────────────────────┘
                          │
                          ▼
┌─────────────────────────────────────────────────────────┐
│ EXECUTION (with MCP)                                    │
├─────────────────────────────────────────────────────────┤
│ "Update the Confluence page at [URL] with these changes"│
│ "Update Aha! feature [ID] status to 'Shipped'"          │
│                                                         │
│ AI makes the updates directly.                          │
│ You review in the actual tool.                          │
└─────────────────────────────────────────────────────────┘
```

Tools: Claude Code + Atlassian MCP (or Aha! MCP if available)

Time Savings: 1-2 hours of manual updates → 10 minutes

Why Demo This:

- Shows AI *in* the workflow, not alongside it
- Practical, weekly task everyone recognizes
- MCP is the "agentic" differentiator
- Good transition to "what's possible with tool integration"

Demo Script: See detailed playbook below.

Workflow 8: Meeting Synthesis

User Story: *"As a PM, I need to synthesize my meeting notes into action items and share them with the team."*

Workflow Sketch:

```
┌─────────────────────────────────────────────────────────┐
│ CAPTURE                                                 │
├─────────────────────────────────────────────────────────┤
│ Options:                                                │
│ • Meeting transcript (Teams, Zoom, Otter)               │
│ • Your handwritten notes (photo → OCR)                  │
│ • Voice memo recorded during/after                      │
└─────────────────────────────────────────────────────────┘
                          │
                          ▼
```

```
┌─────────────────────────────────────────────────────────┐
│ SYNTHESIS PROMPT                                        │
├─────────────────────────────────────────────────────────┤
│ "Synthesize this meeting into:                          │
│                                                         │
│ **Summary** (3–5 sentences max)                         │
│                                                         │
│ **Decisions Made**                                      │
│ - [Decision]: [Rationale]                               │
│                                                         │
│ **Action Items**                                        │
│ - [ ] [Task] — Owner: [Name] — Due: [Date]              │
│                                                         │
│ **Open Questions**                                      │
│ - [Question] — Who needs to answer                      │
│                                                         │
│ **Parking Lot** (topics raised but deferred)"           │
└─────────────────────────────────────────────────────────┘
                          │
                          ▼
┌─────────────────────────────────────────────────────────┐
│ DISTRIBUTION                                            │
├─────────────────────────────────────────────────────────┤
│ • Post to Slack channel                                 │
│ • Create follow-up tasks in Jira (via MCP)              │
│ • Update relevant Confluence pages                      │
└─────────────────────────────────────────────────────────┘
```

Tools: Voice input + Claude.ai (or M365 Copilot in Teams)

Time Savings: 30-45 minutes → 5 minutes

Key Insight: This is Waleed's daily workflow at Daybreak. He uses Claude Code with MCP servers to automatically route meeting outputs to Obsidian, Slack, and Jira.

Workflow 9: AI Feature Evaluation

User Story: *"As a PM, I need to evaluate whether we should add AI to our search feature so I can make a build/buy/skip recommendation."*

Workflow Sketch:

```
┌─────────────────────────────────────────────────────────┐
│ FRAME THE OPPORTUNITY                                   │
├─────────────────────────────────────────────────────────┤
│ "I'm evaluating adding AI to [feature]. Help me think   │
│ through:                                                │
│                                                         │
│ 1. CAPABILITY FIT                                       │
```

```
│    - Is this a task AI is actually good at?                │
│    - What type of AI would this use?                       │
│      (LLM, ML model, rule-based, etc.)                     │
│                                                            │
│ 2. USER VALUE                                              │
│    - What's the user pain point?                           │
│    - How much time/effort does this save them?             │
│    - Would they trust AI for this task?                    │
│                                                            │
│ 3. TECHNICAL FEASIBILITY                                   │
│    - What data do we need?                                 │
│    - What accuracy is required?                            │
│    - How do we handle errors?                              │
│                                                            │
│ 4. BUILD VS BUY                                            │
│    - What vendors offer this?                              │
│    - What's the integration effort?                        │
│    - What's the ongoing cost?"                             │
└────────────────────────────────────────────────────────────┘
            │
            ▼
┌────────────────────────────────────────────────────────────┐
│ RED TEAM                                                   │
├────────────────────────────────────────────────────────────┤
│ "Now argue AGAINST adding AI to this feature.              │
│ What are the strongest reasons to skip this?"              │
└────────────────────────────────────────────────────────────┘
            │
            ▼
┌────────────────────────────────────────────────────────────┐
│ OUTPUT                                                     │
├────────────────────────────────────────────────────────────┤
│ • Recommendation: Build / Buy / Skip                       │
│ • Key assumptions to validate                              │
│ • Risks and mitigations                                    │
│ • Next steps if proceeding                                 │
└────────────────────────────────────────────────────────────┘
```

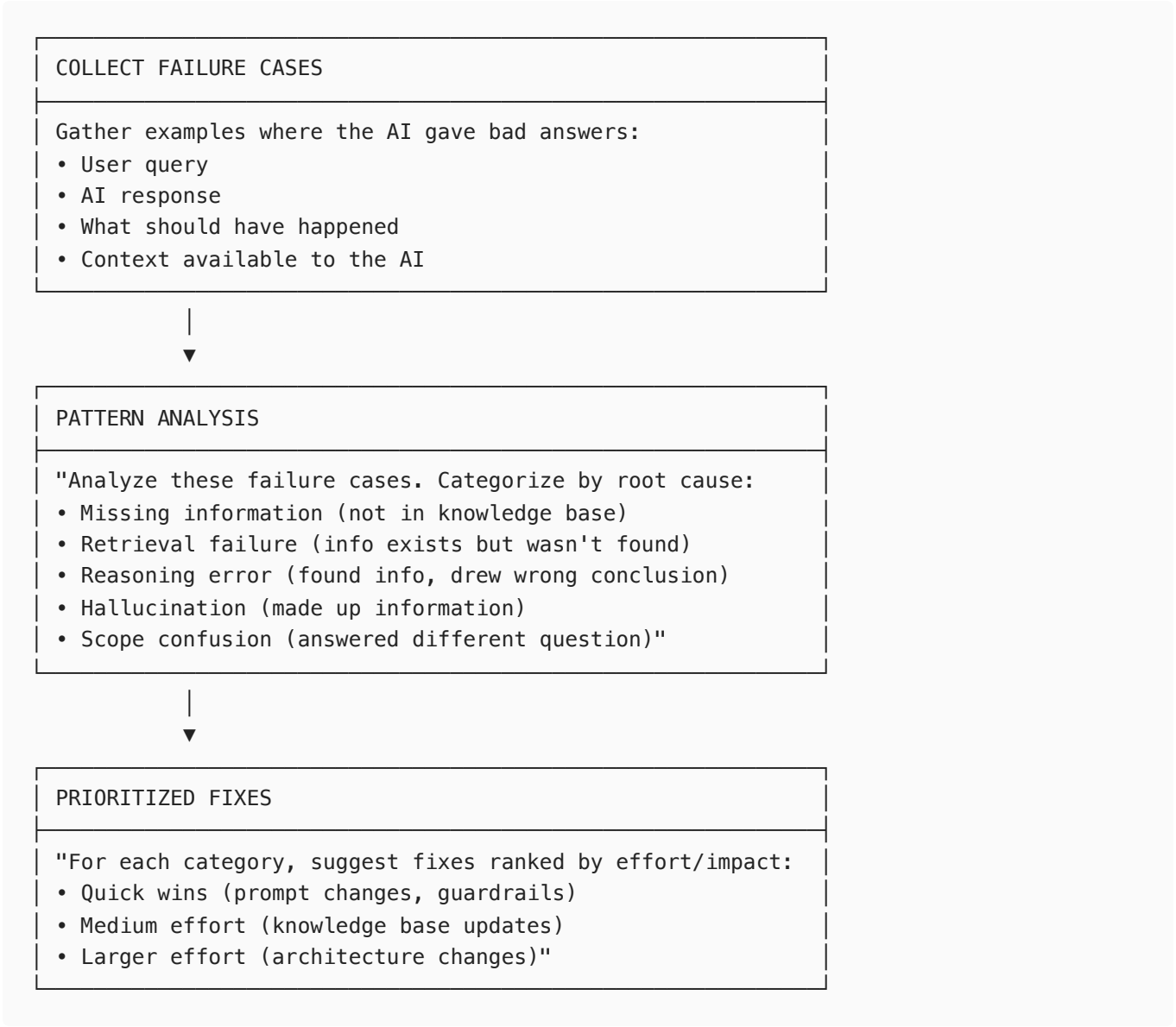
Tools: Claude.ai (reasoning quality matters here)

Key Insight: This framework comes from Teresa Torres's work — treat AI features like any product decision, with explicit hypothesis testing.

Workflow 10: AI Debugging / Evals

User Story: *"As a PM, I need to understand why our RAG bot sometimes gives wrong answers so I can improve it."*

Workflow Sketch:



Tools: Claude.ai + spreadsheet for tracking

Key Insight: This connects to iQmetrix's existing Slack RAG bot. It's a practical way to improve a real system they've built.

From the 9 AI Teams Research:

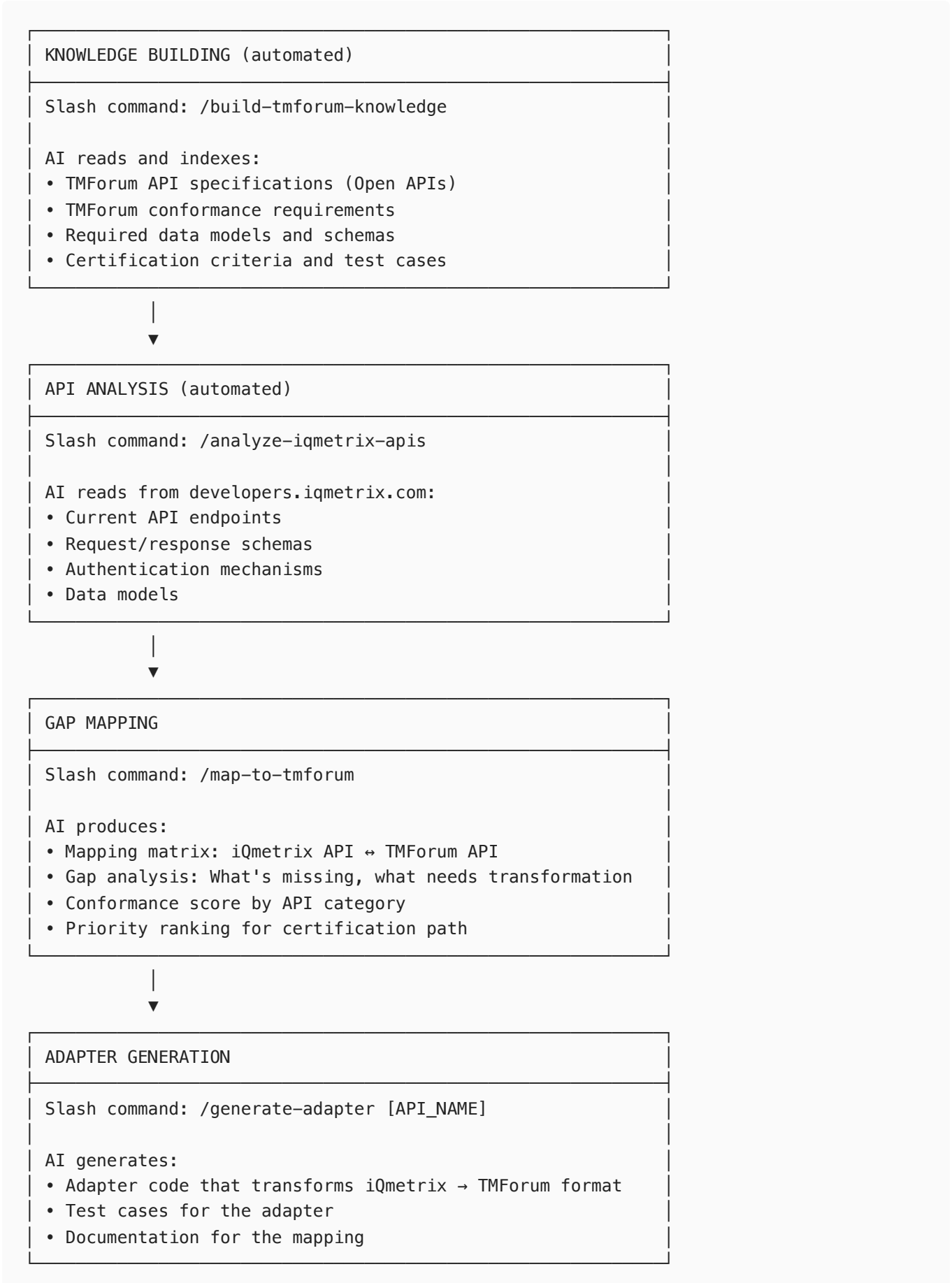
> "Everyone begins with spreadsheets — even Arize, an eval tooling company."

Start simple. Track failures. Look for patterns. Improve iteratively.

Workflow 11: Standards Conformance Automation (TMForum) ⭐ RECOMMENDED DEMO — CAPSTONE

User Story: *"As a PM/technical leader, I need to map our APIs to an industry standard (TMForum) and generate conformance adapters — a certification project we've struggled with for years."*

Workflow Sketch:



Tools: Claude Code with custom slash commands, MCP servers for API access

Time Savings: Months of manual work → Days to weeks

Why Demo This:

- Real business problem iQmetrix has struggled with
- Shows the ceiling of what's possible
- Demonstrates compound workflows (multiple slash commands building on each other)
- "A PM with enough curiosity could have done this" — inspires them to think bigger
- Transitions naturally to "what could YOU automate?"

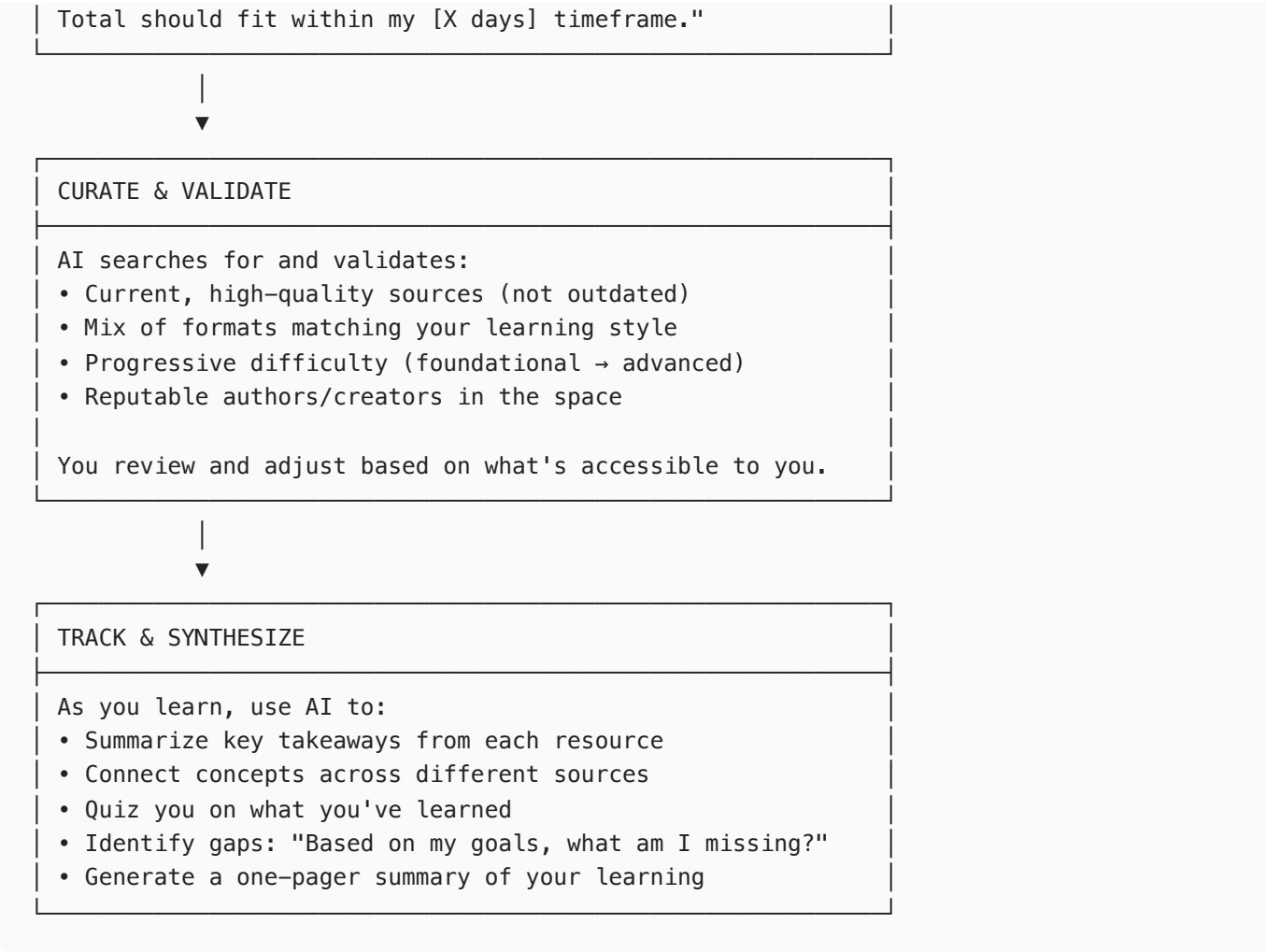
Demo Script: See detailed playbook below.

Workflow 12: Custom Learning Path Generation

User Story: *"As a PM, I need a learning path for a specific topic and I want a custom learning plan that leverages information from all potential sources so I can get to a respectable level in just a few days."*

Workflow Sketch:

```
┌─────────────────────────────────────────────────────────────┐
│ DEFINE LEARNING GOAL                                        │
├─────────────────────────────────────────────────────────────┤
│ Prompt: "I need to learn [TOPIC] well enough to:            │
│ • [Specific goal 1: e.g., lead a stakeholder discussion]    │
│ • [Specific goal 2: e.g., evaluate vendor proposals]        │
│ • [Specific goal 3: e.g., write requirements]               │
│                                                             │
│ Time available: [X days/hours]                              │
│ My current level: [beginner/some familiarity/etc.]          │
│ Learning style preference: [reading/video/hands-on/mixed]"  │
└─────────────────────────────────────────────────────────────┘
                          │
                          ▼
┌─────────────────────────────────────────────────────────────┐
│ AI GENERATES LEARNING PLAN                                  │
├─────────────────────────────────────────────────────────────┤
│ "Create a structured learning plan that includes:           │
│ 1. Core concepts I must understand (priority order)         │
│ 2. Best resources for each concept:                         │
│    - Articles/blogs (with URLs)                             │
│    - YouTube videos or courses                              │
│    - Podcasts or audio content                              │
│    - Books or longer reads (if time permits)                │
│ 3. Hands-on exercises to cement understanding               │
│ 4. Knowledge checks — questions I should be able to answer  │
│ 5. Time estimates for each section                          │
│                                                             │
```



Example Prompt:

> "I need to learn about **API design best practices** well enough to:
>
> - Review our team's API proposals intelligently
> - Have informed conversations with our platform architects
> - Understand trade-offs in REST vs GraphQL decisions
>
> Time available: 3 days (about 2-3 hours per day)
> Current level: I understand what APIs are, but not design principles
> Learning style: Mix of reading and short videos, no long courses
>
> Create a day-by-day learning plan with specific resources."

Tools: Claude.ai or ChatGPT with web search enabled

Time Savings: Weeks of scattered learning → Focused 2-3 day sprint with clear outcomes

Key Insight: The AI doesn't just list resources — it sequences them, estimates time, and creates checkpoints. You're not just learning; you're learning *efficiently* toward a specific goal.

Pro Tips:

- Be specific about your goals — "understand enough to ask good questions" is different from "be able to implement"
- Ask for a "Day 1 checkpoint" — questions you should be able to answer after the first day
- Use the synthesis step to create shareable artifacts (one-pagers, glossaries) that prove your learning

---

PART 3: RECOMMENDED DEMOS & PLAYBOOKS

Demo Selection Rationale

#	Workflow	Why Demo This
2	Interview Synthesis	Foundational, voice input, relatable to all PMs
5	Vibe Coding	Visual wow, shows AI creating real UI
7	MCP Artifact Updates	Shows AI *in* workflow, practical weekly task
11	TMForum Capstone	Shows the ceiling, inspires bigger thinking

Progression: Basic prompting → Voice input → Tool integration → Compound workflows

---

Demo Playbook 1: Interview Synthesis (10 minutes)

Setup Before Session

- Prepare 2-3 fake interview transcripts (or use sanitized real ones)
- Have MacWhisper or Claude mobile app ready
- Have Claude.ai open in browser

Script

[SETUP — 1 min]

> "Let's start with something every PM does: synthesizing customer interviews.
>
> I've got three interview transcripts here — about 45 minutes of conversations with store managers about their inventory challenges.
>
> Normally, I'd spend 2-3 hours reading through these, highlighting themes, organizing quotes. Let me show you a faster way."

[VOICE CAPTURE DEMO — 3 min]

> "First, I'm going to show you voice input. Instead of typing a long prompt, I'll just talk."

Open MacWhisper or Claude mobile

> "Watch this..."

Speak naturally:

> "I just did three interviews with store managers about inventory. The first was with Sarah at the downtown location — she mentioned the inventory count takes her 2 hours every morning and she wishes she could do it from her phone. The second was with Marcus who said his biggest frustration is when the system shows stock that's actually missing. The third was with Jennifer who actually loved the current system but wished it could predict when she'd run low on popular items."

Show the transcription

> "Thirty seconds of talking, and I've captured the key points. Now let's feed this to Claude."

[AI SYNTHESIS — 4 min]

Paste transcription into Claude with this prompt:

> "I conducted 3 customer interviews about inventory management. Here are my notes: [paste]
>
> Synthesize these into:
>
> 1. Top themes by frequency
> 2. Contradictions or tensions between participants
> 3. The most compelling quote for each theme
> 4. Recommended next steps for product"

Show the output

> "In 30 seconds, I have a structured synthesis. Notice it caught that Jennifer contradicts the others — she likes the system. That's the kind of thing you might miss when you're deep in the transcripts."

[KEY TAKEAWAY — 2 min]

> "Two things I want you to notice:
>
> First, voice input. I didn't carefully compose a prompt — I just talked. This is how you brainstorm with AI. Think out loud.
>
> Second, the AI found patterns I might have missed. It's not replacing my judgment — it's giving me a better starting point.
>
> Now imagine doing this with 20 interviews. Same process, same 30 seconds of synthesis. That's the leverage."

Demo Playbook 2: Vibe Coding — Interactive Prototype (12 minutes)

Setup Before Session

- Have v0.dev open (or Lovable/Bolt)
- Prepare iQmetrix brand colors and a screenshot of their current UI
- Have a clear feature idea ready (store manager dashboard)

Script

[SETUP — 1 min]

> "Now let me show you something that changed how I work with designers and engineers.
>
> Imagine you're pitching a new feature — a store manager dashboard. You can describe it in a PRD, sketch it on a whiteboard, or... you can show them a working prototype.
>
> I'm not a designer or developer. But watch this."

[INITIAL GENERATION — 4 min]

Open v0.dev

> "I'm going to describe what I want in plain English."

Type or speak:

> "Create a store manager dashboard for a retail POS system. Include:
>
> - Today's sales total with comparison to yesterday (big number with trend arrow)
> - A list of 5 low-stock items that need attention, each with current quantity and reorder button
> - Today's staff schedule showing who's working which shifts
> - Brand colors: use `#1E3A8A` for primary blue, `#F59E0B` for accents
> - Modern, clean design with rounded corners and subtle shadows"

Submit and wait for generation

> "15 seconds... and I have a working dashboard. Not a mockup — this is real React code that runs in the browser."

Click through the interface

> "The buttons work. The data is sample data but the interactions are real. This is what I can now show my team instead of describing it."

[ITERATION — 4 min]

> "But it's not quite right. The low-stock items should be more prominent. Let me iterate."

Type:

> "Make the low-stock items section more visually urgent — use a warning color background, make items clickable to show a detail modal with reorder options."

Show the update

> "One sentence of feedback, and it adjusted. This is the conversation I used to have over 3 design cycles. Now it's 30 seconds."

[BRAND CONSISTENCY — 2 min]

> "Here's the key insight for product work: I gave it brand colors upfront.
>
> You can go further — upload your design system documentation, screenshots of your current app, even Figma files. The AI maintains consistency.
>
> This isn't about replacing designers. It's about having better conversations with them. Instead of 'I'm imagining something like...' you can say 'Here's a prototype, what would you change?'"

[KEY TAKEAWAY — 1 min]

> "A PM who can prototype is a PM who can communicate. You don't need to learn to code. You need to learn to describe what you want clearly. The AI handles the rest.
>
> Tools like v0, Lovable, and Bolt all do this. Free tiers are generous enough to experiment."

Demo Playbook 3: MCP Artifact Updates (8 minutes)

Setup Before Session

- Have Claude Code running (or show screenshots if setup isn't possible)
- Have a Confluence page open that needs updating
- Prepare a scenario: "Sprint just ended, need to update roadmap status"

Script

[SETUP — 1 min]

> "Now I want to show you what 'AI in your workflow' really means. Not copy-paste. AI that actually updates your tools.
>
> This uses something called MCP — Model Context Protocol. It lets AI tools connect directly to Jira, Confluence, Slack, and other systems.
>
> I'm going to update a Confluence roadmap page without leaving my terminal."

[CONTEXT GATHERING — 2 min]

> "First, let me show Claude what shipped this sprint."

In Claude Code:

> "Check the NSC Jira board for tickets completed in the last 2 weeks. Summarize what shipped."

Show output — list of completed tickets with summaries

> "Without MCP, I'd be clicking through Jira, copying ticket names, writing summaries manually. Claude just did that in 5 seconds."

[ARTIFACT UPDATE — 3 min]

> "Now let's update the roadmap."

Show the current Confluence page

> "This roadmap page has our Q1 features with status. Three items shipped. Let me update them."

In Claude Code:

> "Update the Confluence roadmap page at [URL]. For these three features [list from Jira], change status to 'Shipped' and add the completion date. Add a 'What shipped in Sprint 12' section at the top with bullet point summaries."

Show the updated page

> "Done. The page is updated. I can review it, make manual tweaks if needed, but the grunt work is done."

[KEY TAKEAWAY — 2 min]

> "This is the shift from 'AI as assistant I talk to' to 'AI as agent that takes actions.'
>
> Today you probably use Copilot to draft an email, then copy it into Outlook. With MCP, you'd say 'send this email' and it sends.
>
> The implication for your products: users don't want to copy-paste between AI and their tools. They want AI that operates inside their workflow.
>
> This is what your RAG Slack bot does — it's in Slack, not a separate tool. That's the right instinct."

Demo Playbook 4: TMForum Capstone (10 minutes)

Setup Before Session

- Have the TMForum workflow ready in Claude Code
- Prepare a brief explanation of what TMForum certification is
- Have example output ready (mapping matrix, generated adapter)

Script

[CONTEXT — 2 min]

> "Let me end with something that shows the ceiling of what's possible.
>
> iQmetrix has been working on TMForum certification for a while. TMForum is an industry standard for telecom and retail APIs. Getting certified means enterprise customers can integrate with you more easily.

> The problem: TMForum has hundreds of API specifications. Mapping your APIs to theirs, finding gaps, writing adapters — it's months of tedious work.
>
> A few weeks ago, I built something that changes that."

[KNOWLEDGE BUILDING — 2 min]

> "Step one: teach Claude the TMForum standards."

Show or describe:

> "/build-tmforum-knowledge"

> "This slash command tells Claude to read the TMForum Open API specs, understand their data models, learn the certification requirements.
>
> One command. Runs for about 5 minutes. Now Claude understands TMForum as well as someone who spent weeks studying it."

[API ANALYSIS — 2 min]

> "Step two: analyze the iQmetrix APIs."

Show or describe:

> "/analyze-iqmetrix-apis"

> "This reads from developers.iqmetrix.com. All your public API documentation. Now Claude knows both sides — what TMForum expects, what iQmetrix has."

[MAPPING — 2 min]

> "Step three: create the mapping."

Show the mapping matrix output

> "/map-to-tmforum"

> "In minutes, I have a full mapping matrix. Which iQmetrix APIs correspond to which TMForum APIs. Where the gaps are. What transformations are needed.
>
> This is work that would take a technical team weeks. And the output isn't just analysis — it can generate the actual adapter code."

[KEY TAKEAWAY — 2 min]

> "Here's what I want you to take away:
>
> A PM with enough curiosity could have built this. I'm not an engineer. I described what I wanted, iterated on the workflow, and now I have a reusable system.
>
> Think about the tedious, high-value work in your world. Standards compliance. Vendor evaluations. Data migrations. Documentation updates.

> The question isn't 'can AI do this?' It's 'have I clearly described what I want done?'
>
> That's the PM superpower. You already know how to define requirements. Now you can execute them too."

PART 4: CLOSING & NEXT STEPS

Slide: The PM AI Toolkit (Summary)

SLIDE TITLE: Your AI Toolkit

CONTENT:

For Daily Work:

- Claude.ai / ChatGPT — Research, drafting, analysis
- M365 Copilot — In-context help in Word, Excel, Teams
- Voice input tools — Brainstorming, capture, hands-free work

For Prototyping:

- v0.dev / Lovable / Bolt — Generate working UI from descriptions

For Power Users:

- Claude Code — Agentic workflows, file management, automation
- MCP integrations — Direct connection to Jira, Confluence, Slack

The Progression:

1. Start with prompting basics (this week)
2. Build personal workflows (next month)
3. Connect to your tools (when ready)
4. Automate the tedious stuff (ongoing)

Slide: Evaluating AI for Your Products

SLIDE TITLE: When You're Evaluating AI for iQmetrix Products

CONTENT:

Questions to Ask:

1. **Is this a task AI is actually good at?**
   - Pattern recognition? ✓

- Data synthesis? ✓
- Novel judgment? ✗

2. **What does failure look like?**
   - Annoying but recoverable? → Ship it
   - Costly or trust-breaking? → More guardrails needed
3. **Do users trust AI for this?**
   - High-stakes decisions need human confirmation
   - Routine tasks can be more autonomous
4. **What's the evaluation plan?**
   - Start with spreadsheets
   - Graduate to automated evals
   - Always have a feedback loop

Remember: Domain expertise > AI expertise. Your POS knowledge matters more than knowing how transformers work.

Slide: Next Steps

SLIDE TITLE: Where to Go From Here

CONTENT:

This Week:

- [ ] Try 2-3 workflows from this playbook
- [ ] Set up Claude.ai free account (if not using ChatGPT)
- [ ] Experiment with voice input for brainstorming

This Month:

- [ ] Build one reusable workflow for a recurring task
- [ ] Share what's working with the team

If You Want to Go Deeper:

- Paid Claude Pro seats enable agentic workflows
- Follow-up session on vibe coding for prototyping
- Follow-up session on evaluating AI for iQmetrix products

Resources:

- Teresa Torres on Claude Code: producttalk.org
- "How I AI" podcast with Clare Vo
- This playbook (reference any time)

Slide: Q&A

SLIDE TITLE: Questions?

DISCUSSION PROMPTS:

- Which workflow would save you the most time?
- What's a task you do repeatedly that feels like it could be automated?
- Where do you see AI fitting into iQmetrix products?
- What concerns do you have about adopting AI in your workflow?

APPENDIX: Session Logistics

Recommended Session Structure (90 minutes)

Time	Section	Content
0-20	Preamble	Slides 1-8: Framing, landscape, agents, prompting
20-30	Demo 1	Interview Synthesis (voice input)
30-42	Demo 2	Vibe Coding Prototype
42-50	Demo 3	MCP Artifact Updates
50-60	Demo 4	TMForum Capstone
60-75	Closing	Summary slides, evaluation framework
75-90	Q&A	Discussion, questions, next steps

Condensed Version (60 minutes)

Time	Section	Content
0-15	Preamble	Compressed slides, key concepts only
15-25	Demo 1	Interview Synthesis
25-40	Demo 2	Vibe Coding Prototype
40-50	Demo 3	TMForum Capstone (abbreviated)
50-60	Closing + Q&A	Combined

Pre-Session Checklist

- [ ] Confirm Claude.ai free access is available
- [ ] Test v0.dev / prototyping tool of choice

- [ ] Prepare sanitized interview transcripts
- [ ] Set up voice input tool (MacWhisper, Claude mobile)
- [ ] Have brand colors / style guide ready for vibe coding
- [ ] Prepare TMForum demo outputs (can be screenshots if not live)
- [ ] Test screen sharing and audio setup
- [ ] Send pre-read: "Come with one task you do repeatedly that feels tedious"

Post-Session Materials

- [ ] Share this playbook document
- [ ] Provide links to recommended tools
- [ ] Set up Slack channel for questions / sharing wins
- [ ] Schedule optional follow-up sessions

Document created: December 2025
Advisor: Waleed Ayoub
Partner: Lauren