

Agentic Project Management

User Guide for APM v0.3

CobuterMan

Contents

1	What is APM?	3
1.1	The Problem APM Solves	3
1.2	APM's Approach	3
2	How APM Works	4
2.1	Agents and Their Roles	4
2.1.1	Manager Agent	4
2.1.2	Implementation Agents	4
2.2	You (The User)	4
2.3	Key Components	5
2.3.1	Implementation Plan	5
2.3.2	Memory Bank	5
2.3.3	Handover Protocol	5
2.4	The Workflow Loop	6
2.4.1	APM Workflow Diagram	7
3	Starting Your APM Session	8
3.1	Prerequisites	8
3.2	Setting Up APM Assets	8
3.2.1	Option 1: Clone the Repository	8
3.2.2	Option 2: Use the APM Template (Recommended)	8
3.2.3	Option 3: Manual Access	8
3.3	Initializing Your Manager Agent	9
3.4	Collaborative Context Gathering	10
3.4.1	Project Overview Discussion	10
3.4.2	Discovery Process Options	11
3.5	Implementation Plan and Memory Bank Creation	12

3.5.1	The Planning Phase	12
3.5.2	File Creation and Review	13
4	Working with Implementation Agents	14
4.1	The Task Assignment Process	14
4.1.1	Issuing your first Task Assignment Prompt	14
4.1.2	Understanding the Task Assignment Prompt Structure	15
4.2	Onboarding Implementation Agents	15
4.2.1	Creating Agent Sessions	15
4.3	Task Execution Workflow	15
4.3.1	Delivering Task Assignments	15
4.3.2	Agent Execution	16
4.4	Memory Bank Logging	17
4.4.1	Confirming and Instructing Logging	17
4.4.2	Logging Format	17
4.5	Manager Review and Next Steps	18
4.5.1	Reporting to Manager	18
4.5.2	Manager Evaluation	18
4.5.3	Workflow Continuation	19
5	Understanding and Managing Handovers	19
5.1	When to Consider Handovers	19
5.1.1	Context Window Awareness	19
5.1.2	Performance Warning Signs	20
5.2	The Handover Process	21
5.2.1	Initiating a Handover	21
5.2.2	Handover Artifacts	22
5.3	Executing the Handover	23
5.3.1	Starting the New Agent	23
5.3.2	Verification and Confirmation	23
5.3.3	Resuming Operations	24
6	Summary	25
6.1	Key Takeaways	25
6.2	Getting Started	25

1 What is APM?

Agentic Project Management (APM) is a multi-agent workflow for managing complex projects within an AI IDE environment. It applies real project management principles to AI workflows like task breakdown, role assignment, progress tracking, and seamless handoffs between team members.

APM utilizes chat sessions in your AI IDE as separate agent instances, each with its own context scope and memory. This allows for more focused interactions and reduces the cognitive load on any single agent.

Dividing the project workload among multiple agents minimizes the risk of context loss and reduces hallucinations, leading to more reliable and consistent results.

1.1 The Problem APM Solves

Working with AI assistants on large projects often feels chaotic. You start with a simple question, and before you know it, you're 50 messages deep in a conversation where the AI has forgotten your original requirements, contradicts its earlier suggestions, or starts hallucinating details that never existed.

These issues arise from fundamental limitations of LLMs: **Context Window Limits**.

This constraint feels "heavier" within AI IDEs, when often times Context Windows are shrunk even further to maintain profitable interactions with the model's provider. As conversations grow, the AI struggles to keep track of everything, leading to confusion, errors, and wasted time.

APM addresses these issues by providing a clear framework that prioritizes structured interaction, explicit context management and integration, and efficient, targeted communication with AI assistants through meta-prompting.

1.2 APM's Approach

Instead of fighting these limitations, APM works around them. The framework uses multiple AI agent instances, each with a specific role and clear responsibilities, coordinated through structured protocols and persistent documentation. The result is a workflow that feels more like working with a well organized team than wrestling with a single overloaded AI assistant.

Real World Analogy: Think of APM like running a software development team. You have a project manager who understands the big picture, developers who focus on specific tasks, and clear documentation that keeps everyone aligned. Developers document each task execution, and the manager reviews the logs for coordination. The difference is that your "team members" are AI assistants in separate chat sessions.

2 How APM Works

APM employs a multi-agent system built around two agent types and the User, utilizing well established prompt and context engineering techniques as its main workflow components.

2.1 Agents and Their Roles

2.1.1 Manager Agent

Your central coordinator. The Manager Agent is responsible for:

- Understanding your project goals and requirements
- Creating a detailed Implementation Plan that breaks down your project into manageable phases, tasks and sub-tasks
- Issuing task assignment prompts for Implementation Agents to execute
- Reviewing completed work and deciding what comes next
- Maintaining the overall project context and ensuring nothing gets lost
- Managing handovers when agents approach their context limits

2.1.2 Implementation Agents

Your workers who execute specific parts of the project. Each Implementation Agent executes tasks assigned to it by the Manager Agent. Their responsibilities include:

- Executing tasks as per the Implementation Plan, issued by the Manager Agent (e.g. Coding, Debugging, Research)
- Communicating with the User for clarifications, collaboration needs, and additional context
- Logging their execution attempt and task status in the Memory Bank for the Manager Agent to review

2.2 You (The User)

You serve as the crucial communication bridge between all agents. AI IDEs don't offer pipelines for Agent-to-Agent communication. APM relies on you to facilitate this communication and supervise the development. Your responsibilities include:

- Defining project goals and providing requirements to the Manager Agent
- Reviewing and approving the Implementation Plan created by the Manager Agent
- Assigning tasks to Implementation Agents by delivering task assignment prompts from the Manager Agent
- Providing feedback and guidance to Implementation Agents during execution
- Making key decisions about project direction in collaboration with the Manager Agent

2.3 Key Components

2.3.1 Implementation Plan

A detailed, structured document (default `Implementation_Plan.md`) that serves as your project blueprint. Created by the Manager Agent after understanding your goals and requirements, it breaks down your project into:

- **Phases:** Logical groupings of related work
- **Tasks:** Specific units of work assigned to individual agents
- **Sub-tasks:** Granular steps within each task
- **Dependencies:** Clear relationships between different pieces of work

2.3.2 Memory Bank

Your project's persistent memory system. APM uses the traditional single-file approach for simple projects, stored in `Memory_Bank.md` or a dynamic structure across multiple directories and files, stored in a `Memory/` root directory for complex projects. Implementation Agents log their work here in a standardized format, creating:

- A chronological record of all project activities
- Context for Manager Agent to review completed work and make informed decisions
- An auditable history of decisions and changes throughout the project lifecycle
- Documentation of issues and their resolutions for future reference and adaptation
- A foundation for handovers when agents reach their context limits

2.3.3 Handover Protocol

A well-defined procedure for transferring project context and responsibilities between agent instances. This becomes essential when chat sessions (agents) approach their context limits or when you need to switch focus. The protocol uses two key artifacts:

- **Handover File:** A comprehensive context dump from the outgoing agent instance containing current active context
- **Handover Prompt:** Initialization instructions for the replacement agent instance to recover context and resume work

2.4 The Workflow Loop

After project planning, the workflow follows a predictable cycle per Implementation Plan task. Here is a detailed breakdown of the workflow loop:

1. **Manager Issues Task:** Creates a task assignment prompt for the next piece of work
2. **User Delivers:** Transfers the meta-prompt to the appropriate Implementation Agent chat session via copy-paste
3. **Implementation Agent Executes:** Implementation Agent executes the assigned task (either successfully or with issues/blockers)
4. **Implementation Agent Reports to User:** Reports status, outputs, and any issues back to you
5. **User Confirms:** Reviews the work and confirms or reiterates with modification requests
6. **Implementation Agent Logs:** Implementation Agent proceeds to log the work in the Memory Bank after explicit User confirmation
7. **User Reports to Manager:** Informs the Manager Agent of the completed work and logs by the Implementation Agent
8. **Manager Reviews:** Evaluates the logged work and determines next steps
9. **Cycle Repeats:** Process continues with the next task or addresses any issues

This loop continues until your project is complete, with handovers occurring as needed when agents approach their context limits.

Task Loop rationale: Each implementation agent maintains focus on their contextual scope without getting overwhelmed by the full project context. The Manager Agent sees the big picture but does not execute, Implementation Agents dive deep into their tasks but don't make project decisions, and the Memory Bank ensures nothing gets lost along the way.

2.4.1 APM Workflow Diagram

To help you understand the workflow loop, below is a diagram illustrating the full APM v0.3 session, from the Planning Phase through to Implementation, showing all major components and their interactions.

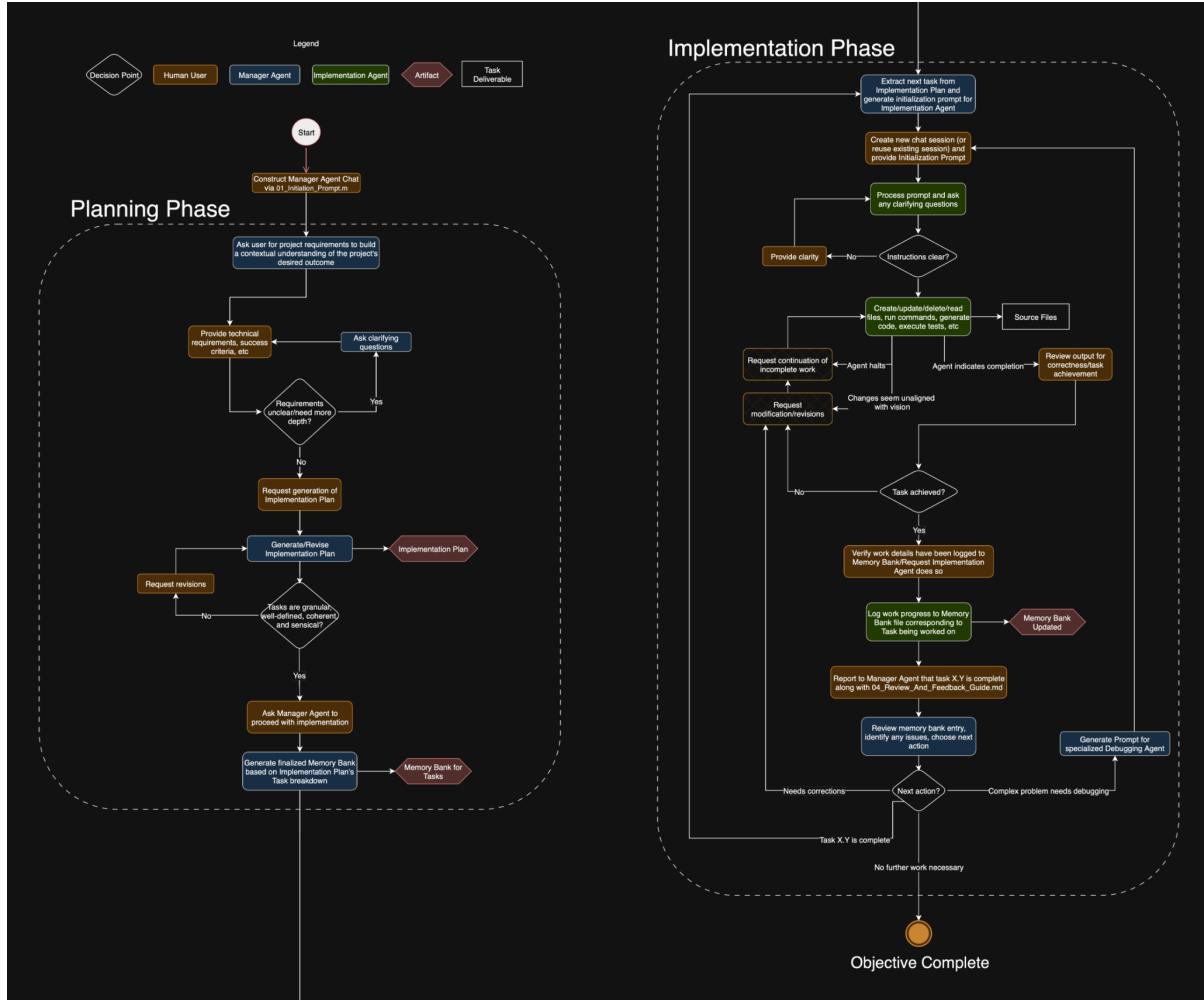


Figure 1: APM v0.3 Workflow. Diagram generously created and contributed by [@cachemap](#).

The diagram illustrates the communication between the Agents, the Users actions, and the key artifacts generated throughout the workflow. You can trace how tasks and information flow between the Manager Agent, Implementation Agents, through Task Assignment Prompts and the Memory Bank, as well as where the User steps in to coordinate, confirm, or transfer context.

Tip: Whenever you feel lost during an APM session, reference this guide and use the workflow diagram as a complementary visualization. The diagram provides a clear visual understanding of how APM orchestrates collaboration, decision-making, and documentation at every stage of the project.

3 Starting Your APM Session

Getting started with APM involves setting up your project structure and initializing your Manager Agent. This process typically takes 15-30 minutes, depending on your project's complexity, but creates a solid foundation for efficient project execution.

3.1 Prerequisites

Before diving in, make sure you have:

- **AI IDE Platform:** Access to an AI IDE platform where you can maintain multiple separate chat sessions with file operation tools (VS Code, Cursor, Windsurf etc.)
- **Workspace:** A dedicated folder or directory for your project files
- **APM Assets:** The APM prompts and guides from the GitHub repository or a template, cloned into your project's workspace
- **Project Idea:** A project idea complex enough to benefit from structured management

3.2 Setting Up APM Assets

You have three options for accessing the APM framework materials:

3.2.1 Option 1: Clone the Repository

Clone the APM repository directly into your project workspace:

```
git clone https://github.com/sdi2200262/agentic-project-management
```

This gives you direct access to all prompts, guides, and format definitions, plus easy updates to newer versions.

3.2.2 Option 2: Use the APM Template (Recommended)

Click "Use this template" on the GitHub repository to create your own copy with the entire APM structure pre-filled. Clone the template repo into your project's workspace as you would with the original repository. This is ideal if you want to customize the framework for your specific needs.

3.2.3 Option 3: Manual Access

For basic usage, you can copy-paste prompt contents directly from the GitHub repository as needed. This requires more manual effort but works if you prefer not to mess with Git.

3.3 Initializing Your Manager Agent

Activate the first Manager Agent instance to kick off your APM session. This agent will guide you through the project setup and planning process.

1. **Open a New Chat Session:** Create a dedicated, clean chat session in your AI assistant platform. Name it clearly (e.g., "Manager Agent").
2. **Provide the Initiation Prompt:** Either as context or by navigating to `prompts/00_Initial_Manager_Setup/01_Initiation_Prompt.md` in your APM assets. Copy the entire content and paste it as the very first message to your Manager Agent.
3. **Confirm Asset Availability:** Your Manager Agent will ask about APM asset availability. Respond based on your setup choice from the previous section.

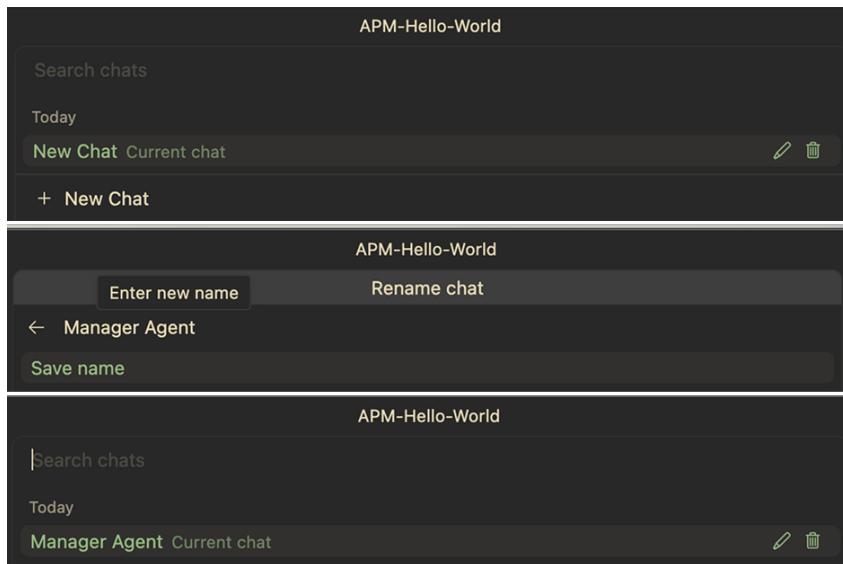


Figure 2: Opening a new AI chat session and naming it "Manager Agent"

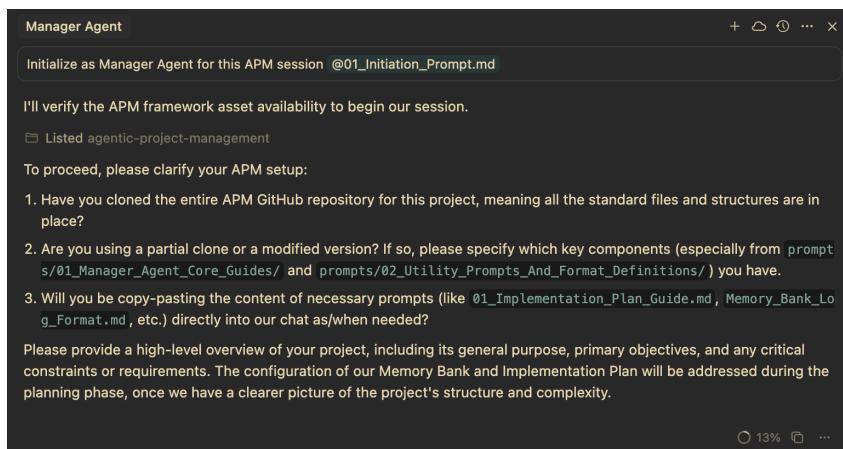


Figure 3: Initializing a Manager Agent instance in Cursor by providing the initiation prompt

3.4 Collaborative Context Gathering

Now comes the crucial phase where you work with your Manager Agent to build a comprehensive contextual understanding of your project. This collaborative process ensures your Implementation Plan will be accurate and complete. Provide as much detailed information as possible to set your project up for success.

3.4.1 Project Overview Discussion

Your Manager Agent will guide you through understanding your project from multiple angles:

- **High-level Goals:** What are you trying to accomplish?
- **Already Existing Assets:** Do you have any existing code, documentation, or designs?
- **Technical Requirements:** What technologies, platforms, or constraints are involved?
- **Scope and Scale:** How complex is this project? What are the major components?
- **Timeline and Priorities:** Are there deadlines or critical path items?
- **Success Criteria:** How will you know when the project is complete?

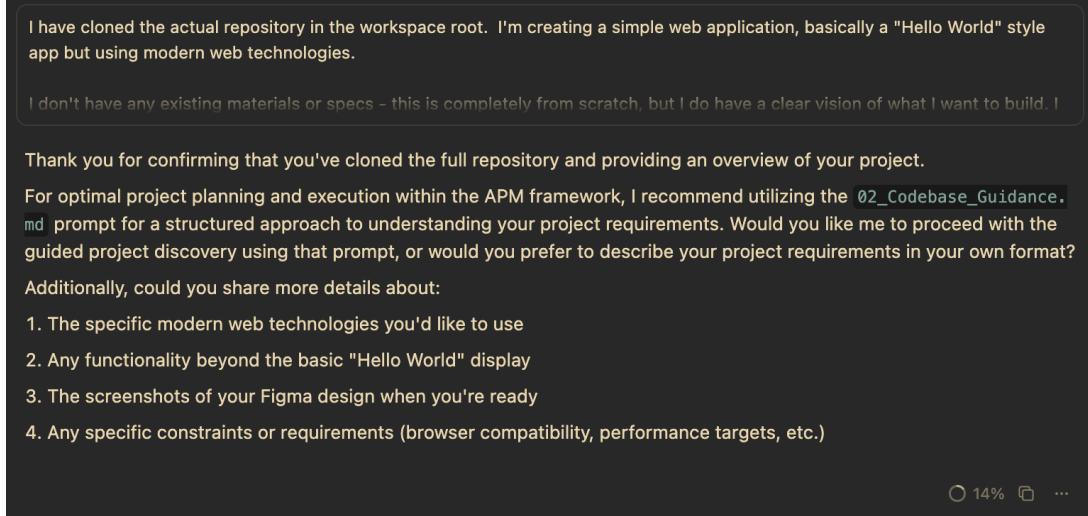


Figure 4: Manager Agent guiding the user through project discovery and context gathering

3.4.2 Discovery Process Options

The Manager Agent will offer you two approaches for gathering detailed context:

- **User-Directed Discovery:** You provide information in your own structure and timeline
- **Manager-Guided Discovery (Recommended):** The Manager Agent leads you through a systematic and adaptive questionnaire

Choose the approach that feels more natural for your project and communication style.

The Manager-Guided Discovery is generally more efficient, especially for complex projects, as the conversation is guided through a framework which the LLM references to ensure all relevant aspects are covered.

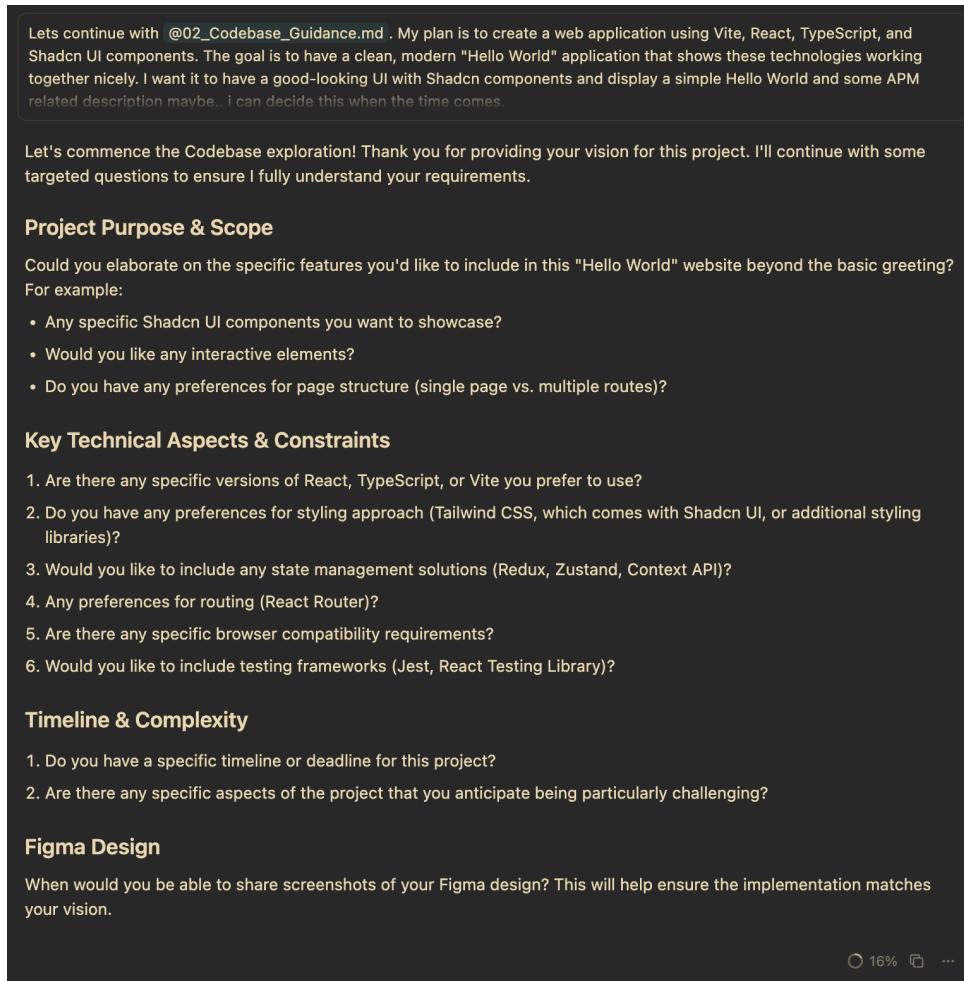


Figure 5: Manager Agent guiding the user through a structured discovery process

For best results: Don't rush this phase. The time you invest in thorough context gathering pays dividends later with more accurate task assignments, no context gaps, and fewer misunderstandings. If you have existing documentation, requirements, or code, mention it early so the Manager Agent can incorporate it into the plan.

3.5 Implementation Plan and Memory Bank Creation

Once your Manager Agent has comprehensive context, it will either autonomously determine when it's ready to create your Implementation Plan and initialize the Memory Bank or it will present a summary and ask for your confirmation to proceed. This typically happens after several rounds of discussion and clarification.

3.5.1 The Planning Phase

The Manager Agent will present you with:

- **High-level Summary:** An overview of how it plans to structure your project
- **Memory Bank Strategy:** Whether to use a single file or multiple organized files
- **Phase Breakdown:** How the work will be organized into logical phases
- **Agent Assignments:** How the task workload will be distributed among Implementation Agents

The screenshot shows a dark-themed application window with white text. At the top, a message says "Please proceed, this looks accurate." Below it, a confirmation message reads: "Thank you for the confirmation. I believe I now have a sufficient and comprehensive understanding of the project requirements, scope, and technical context from our guided discovery. I am now ready to proceed to **Phase B: Strategic Planning & Implementation Plan Development**, as outlined in my primary initiation protocol." It lists two items: "④ Read 02_Memory_Bank_Guide.md" and "④ Read 01_Implementation_Plan_Guide.md". A note follows: "Based on our discussion and the information gathered, I have formulated a high-level strategic plan to achieve the project objectives. Here is an overview:"

High-Level Implementation Plan Summary:
The project will be structured in 3 main phases:
1. **Phase 1: Project Foundation & Setup** - Setting up the development environment, initializing the Vite + React + TypeScript project, and configuring Shadcn UI with Tailwind CSS
2. **Phase 2: Core Component Development** - Building the main UI components including the floating header, hero section with APM workflow diagram, and interactive elements
3. **Phase 3: Content Integration & Polish** - Implementing the remaining sections (features showcase, IDE integration, steps guide), adding animations, and final responsive design adjustments

The plan will involve 3 specialized Implementation Agents:
• **Agent_Setup_Specialist**: Handles project initialization and tooling configuration
• **Agent_Frontend_Dev**: Develops React components and implements the Figma design
• **Agent_Polish_Specialist**: Focuses on animations, responsiveness, and final optimizations

Memory Bank Structure Proposal & Justification:
Following the `02_Memory_Bank_Guide.md`, and considering the project's moderate complexity with multiple distinct phases, specialized agents, and various component development tasks, I recommend a multi-file directory system (`Memory/`) with subdirectories corresponding to each phase. This structure will effectively organize logs for the different workstreams and agents, making it easier to track progress across the distinct development phases.

Proceeding to Implementation Plan Creation:
I am now proceeding to create the `Implementation_Plan.md` file. This document will contain the detailed breakdown of phases, tasks, sub-tasks, dependencies, and agent assignments based on the overview I just provided. I will use the standard format defined in `prompts/01_Manager_Agent_Core_Guides/01_Implementation_Plan_Guide.md`.
I am also proceeding to create the necessary Memory Bank directory structure based on the multi-file system I've proposed, following the detailed setup instructions outlined in `prompts/01_Manager_Agent_Core_Guides/02_Memory_Bank_Guide.md`. This will involve creating the `/Memory/` directory, its `README.md`, and initial phase subdirectories with task log files, initialized as per that guide.

Implementation_Plan.md (new) +173 *
Implementation Plan

Project Goal: Develop a modern APM (Agentic Project Management) landing page website using Vite, React, TypeScript, and Shadcn UI to demonstrate both 1
Phase 1: Project Foundation & Setup – Agent Group Alpha (Agent_Setup_Specialist)

Figure 6: Manager Agent guiding the user through a structured planning process

3.5.2 File Creation and Review

The Manager Agent will now create the actual project files:

- **Implementation_Plan.md:** Your detailed project blueprint
- **Memory Bank files:** The logging structure for your project

Review these files, suggest modifications until you're satisfied with the setup.

The screenshot shows a terminal window with four tabs, each displaying a file or log entry. The tabs are:

- Implementation_Plan.md (new) +173 ***: Contains the detailed project blueprint.
- README.md (new) +14 ***: Contains information about the APM Project Memory Bank Directory.
- Task_1.1_Dev_Environment_Init_Log.md (new) +13 ***: Contains the log for Development Environment Initialization.
- Task_1.2_Asset_Prep_Basic_Styling_Log.md (new) +13 ***: Contains the log for Asset Preparation & Basic Styling.
- Task_2.1_Navigation_Header_Component_Log.md (new) +13 ***: Contains the log for Navigation Header Component.
- Task_2.2_Hero_Section_APM_Workflow_Log.md (new) +13 ***: Contains the log for Hero Section with APM Workflow Diagram.

At the bottom of the terminal window, there are buttons for 'Undo All' and 'Keep All'.

Figure 7: Implementation Plan and Memory Bank files created by the Manager Agent

Quality Check: Before approving your Implementation Plan, verify that tasks are granular, well defined, and include explicit agent assignments, clear dependencies between related work, and sufficient detail for each task to be executed independently. A well-crafted plan saves significant time during execution.

4 Working with Implementation Agents

Now that you have your Implementation Plan, it's time to start executing tasks. This is where Implementation Agents come into play, each focusing on specific aspects of your project while the Manager Agent coordinates the overall workflow.

4.1 The Task Assignment Process

Every piece of work in APM starts with a carefully crafted Task Assignment Prompt. These are meta-prompts created by the Manager Agent that provide all the necessary context and instructions for an Implementation Agent to execute a specific task.

4.1.1 Issuing your first Task Assignment Prompt

Working with your Manager Agent to create the first prompt is straightforward. Provide the `03_Task_Assignment_Prompts_Guide.md` file to your Manager Agent and ask:

"Give me the task assignment prompt for the first task in the Implementation Plan."

Tip 1: You only have to provide the guide once during the Manager Agent's session, until it reaches its context limit. After Handover procedures, you will need to provide the guide again to the new Manager Agent instance.

Tip 2: Sometimes the Manager Agent will create the meta-prompt in plain text, and not in a markdown code block as defined in the guide. This depends on the specific AI IDE platform you're using. If this happens, you can simply clarify this to the Manager Agent by saying:

```
Your planning looks correct. Read the @03_Task_Assignment_Prompts_Guide.md and give me the Task Assignment Prompt for the first task in the Implementation Plan
please provide the task assignment prompts in markdown code blocks from now on. Give me the task assignment prompt for task 1 once more in markdown code block

# markdown
# APM Task Assignment: Development Environment Initialization
## 1. Agent Role & APM Context
**Introduction:** You are activated as an Implementation Agent within the Agentic Project Management (APM) framework for the APM Landing Page Website project.
**Your Role:** As an Implementation Agent, your core function is to execute specific tasks assigned to you based on our detailed project plan, performing necessary actions and monitoring progress.
**Workflow:** You interact directly with the User, who acts as the communication bridge with the Manager Agent. You will report your progress, results, or any issues back to the Manager Agent.
**Note:** For a more detailed explanation of your role and the APM workflow, please refer to `prompts/02_Utility_Prompts_And_Format_Definitions/Implementation_Agent_Onboarding.md`.

## 2. Task Assignment
**Reference Implementation Plan:** This assignment corresponds to 'Phase 1, Task 1.1 - Agent_Setup_Specialist: Development Environment Initialization' in the Implementation Plan.
**Objective:** Set up the complete development environment with Vite, React, TypeScript, and prepare for Shadcn UI integration.
**Detailed Action Steps:**
1. **Initialize Vite project with React and TypeScript template:**
```

Figure 8: Clarifying the Task Assignment Prompt format with the Manager Agent

4.1.2 Understanding the Task Assignment Prompt Structure

A well-crafted Task Assignment Prompt from the Manager Agent should include the following elements:

- **Task Reference:** Clear identification from the Implementation Plan
- **Context:** Relevant background information and dependencies
- **Objective:** What needs to be accomplished
- **Detailed Instructions:** Step-by-step guidance for execution
- **Expected Output:** Specific deliverables and success criteria
- **Logging Instructions:** Where and how to record the work

4.2 Onboarding Implementation Agents

Each Implementation Agent needs proper initialization before it can start working effectively. This onboarding process sets up their understanding of the APM framework and their role within your project.

4.2.1 Creating Agent Sessions

Similarly to the previous section, you will create dedicated chat sessions for each Implementation Agent. This ensures that each agent has focused context scope for its tasks.

1. **Open a New Chat Session:** Create a separate, dedicated chat session for each Implementation Agent
2. **Name It Clearly:** Use the name described in the Implementation Plan like "Agent_Setup_Specialist" or "Agent_Frontend"
3. **Provide Onboarding:** Provide as context or copy the entire content of `Implementation_Agent_Onboarding.md` and paste it as the first message

4.3 Task Execution Workflow

Once your Implementation Agent is onboarded, the actual work execution follows a predictable pattern that maintains quality and accountability.

4.3.1 Delivering Task Assignments

1. **Copy the Prompt:** Take the Task Assignment Prompt prepared by your Manager Agent
2. **Paste to Agent:** Deliver it to the appropriate Implementation Agent's chat session
3. **Clarify if Needed:** Answer any clarifying questions the agent might have

4.3.2 Agent Execution

Your Implementation Agent will now work on the assigned task. During execution, the agent might:

- Ask clarifying questions to ensure understanding
 - Request additional context or resources
 - Report intermediate progress for complex tasks
 - Identify dependencies or blockers that need attention

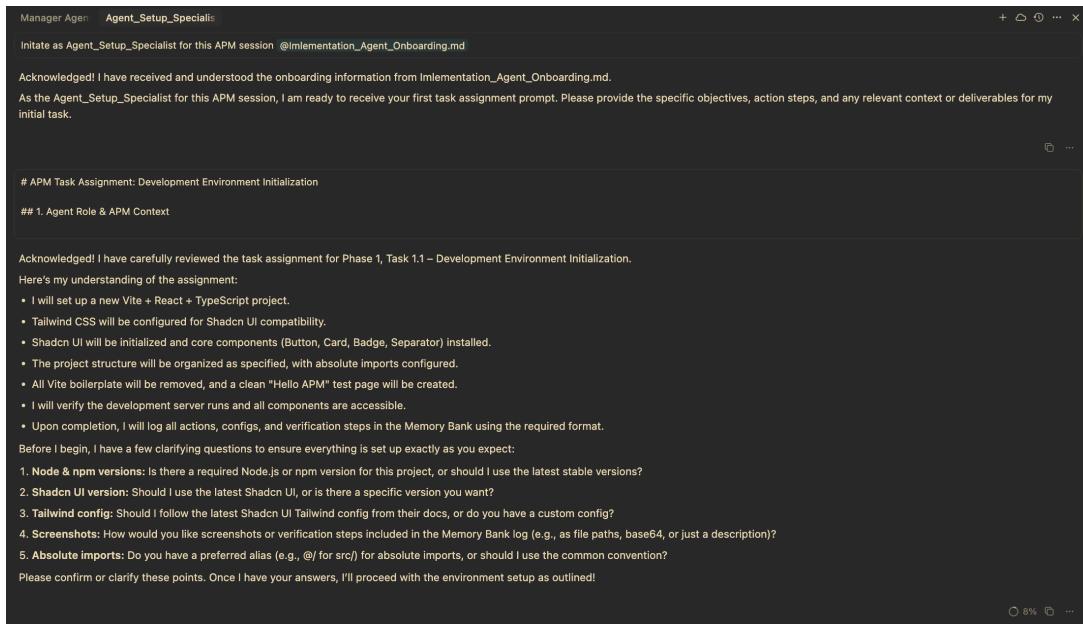


Figure 9: Implementation Agent receiving task and asking clarifying questions

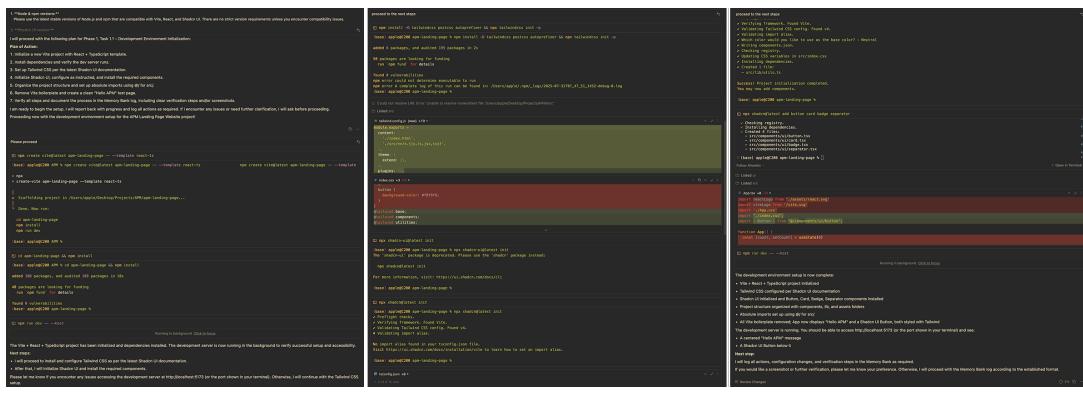


Figure 10: Implementation Agent executing the task and providing progress updates

After executing the task, either successfully or with issues/blockers, the Implementation Agent will report back to you with its findings and results. It will ask confirmation to log its work in the Memory Bank, which is a crucial step for maintaining project continuity. If you are happy with the task execution, you can instruct the agent to log its work, otherwise you can ask it to make modifications or corrections.

4.4 Memory Bank Logging

Consistent logging is what keeps your project organized and ensures no work gets lost. This step is crucial for maintaining project continuity and enabling effective handovers.

For the first memory log entry of an Implementation Agent instance, you will have to provide the

`Memory_Bank_Logging_Format.md` which contains the standardized format for logging work, just like you did with the first Task Assignment Prompt of the Manager Agent.

4.4.1 Confirming and Instructing Logging

Reviewing an Implementation Agent's work:

1. **Review the Output:** Examine what the agent accomplished and any issues reported
2. **Decide on Logging:** Determine if the work should be logged based on task completion and quality
3. **Instruct the Agent:** Give clear direction like "Proceed to log this work to your designated log file for this task, using the standard format"

The screenshot shows a code editor window with a dark theme. At the top, there is a message: "Please proceed to log in your designated log file for this task, following the `@Memory_Bank_Log_Format.md` .". Below this, there is a link: "Read Task_1.1_Dev_Environment_Init_Log.md". The main content area displays a log entry from "Task_1.1_Dev_Environment_Init_Log.md +62". The log entry includes sections for "Agent", "Task Reference", "Summary", and "Details". The "Summary" section states: "Initialized the APM Landing Page development environment with Vite, React, TypeScript, Tailwind CSS, and Shadcn UI. Verified that the dev server runs and all imports are resolved." The "Details" section is partially visible. At the bottom of the log entry, it says: "The log entry for Phase 1 / Task 1.1 (Development Environment Initialization) has been successfully added to the designated Memory Bank file, following the required format." Below the log entry, there is a summary: "Summary of the log:" followed by a bulleted list: "All setup steps (Vite, React, TypeScript, Tailwind, Shadcn UI, project structure, absolute imports) are documented.", "Key configuration/code snippets and verification steps are included.", and "Status is marked as Completed, with no issues or blockers.". There is also a note: "If you need any further details, screenshots, or wish to proceed to the next task, please let me know!". At the bottom left, there is a "Review Changes" button. On the right side, there are several small icons for navigating the code editor.

Figure 11: First Implementation Agent log work after execution

4.4.2 Logging Format

Implementation Agents log their work using a standardized format that includes:

- **Header:** Agent identification and task reference
- **Status:** Completion status (Completed, Partial, Blocked, Error)
- **Output:** Deliverables and key results
- **Issues:** Problems encountered or none
- **Next Steps:** Follow-up actions or none

4.5 Manager Review and Next Steps

After an Implementation Agent logs its work, you return to your Manager Agent chat session to close the task loop and determine what happens next.

For the first task, you will have to provide the `04_Review_And_Feedback_Guide.md` to your Manager Agent, which contains the framework for reviewing completed tasks, similarly as we did with the Task Assignment Prompt and Memory Bank Logging Format.

4.5.1 Reporting to Manager

Inform it of the completed work: "Agent A has completed Task X.Y and logged the results to `Task_X.Y_Task_Name_Log.md` in the Memory Bank."

4.5.2 Manager Evaluation

Your Manager Agent will then:

- Review the logged work for quality and completeness
- Assess whether the task objectives were met
- Identify any issues that need correction
- Determine the next appropriate action

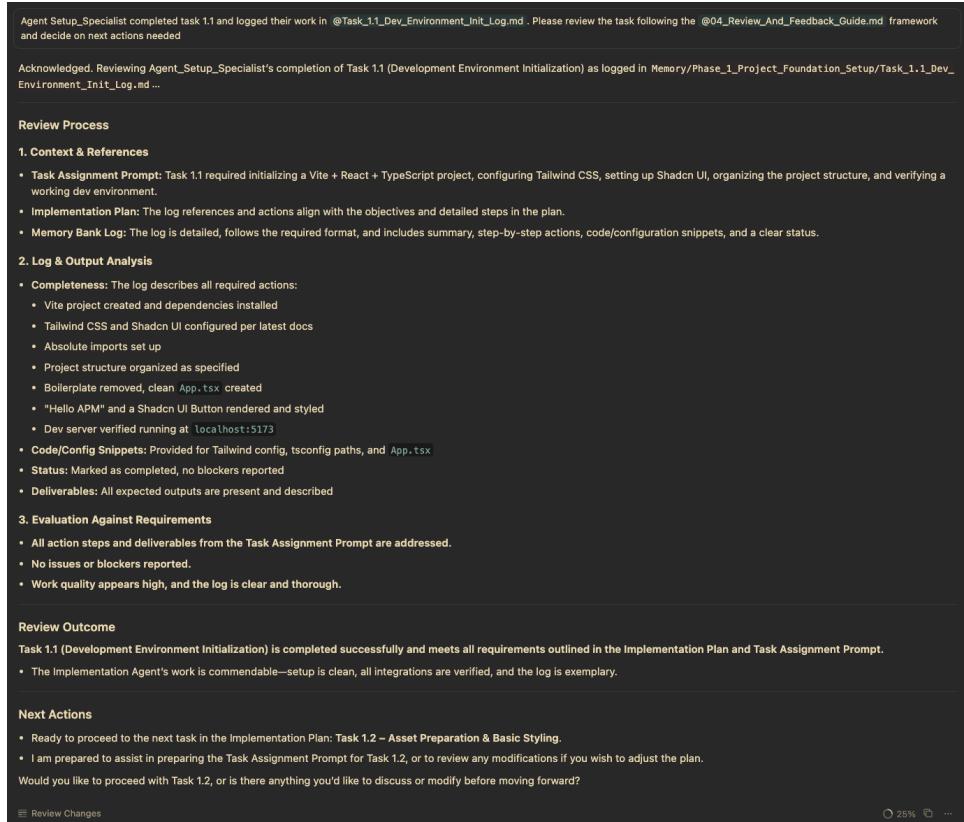


Figure 12: Manager Agent reviewing the logged work of an Implementation Agent and deciding next steps

4.5.3 Workflow Continuation

Based on its review, the Manager Agent will either:

- **Continue Forward:** Prepare the next task assignment prompt
- **Request Corrections:** Create a follow-up prompt to address issues
- **Assign Debugging:** Engage a specialized agent for complex problems
- **Update the Plan:** Modify the Implementation Plan if requirements have changed

This cycle continues, with each completed task building toward your project goals while maintaining clear documentation and accountability throughout the process.

You may make comments at any critical review step, to collaborate with the Manager Agent and provide additional context, clarification or make project decisions.

Efficiency Tip: After the first review, when the Review and Feedback Guide is in active context of the Agent, you may ask the Manager to issue the Next Task Assignment prompt in the same response, if you are certain that the previous task was completed successfully and no issues were reported.

This saves time, premium requests from your subscription and keeps the workflow moving smoothly. For example ask:

"Agent A has completed Task X.Y and logged the results to Task_X.Y_Task_Name_Log.md in the Memory Bank. Please review their work and if all is well issue the next task assignment prompt for Task X.Z."

5 Understanding and Managing Handovers

As your project progresses and conversations with agents grow longer, you'll eventually encounter the context window limit of this chat session. This is where APM's Handover Protocol becomes essential, allowing you to seamlessly transfer project responsibilities to fresh agent instances without losing context.

5.1 When to Consider Handovers

Recognizing the right time for a handover is crucial for maintaining project quality and efficiency. **Don't wait until an agent completely fails; be proactive about the signs.** If a handover is done late, the agent will likely have already started to hallucinate which will pollute the context of the new agent instance.

5.1.1 Context Window Awareness

Think of an AI agent's context window like active memory. As conversations grow longer, the agent has to "work harder" to keep track of everything that's been discussed or done.

Some AI assistants typically start showing degraded performance well before hitting their absolute limits.

Also remember that AI IDEs often shrink context windows even further to maintain profitable interactions with the model's provider, so it's even more important to be proactive about handovers.

A helpful analogy is Cursor's context window visualization (see Figure 12 bottom right). When you see the context usage climbing above 80-90%, it's time to start thinking about a handover, even if the agent seems to be performing well.

Other AI IDE platforms have similar indicators, it's up to you to familiarize yourself with the specific platform you are using.

Some may not have a visual indicator, but you can still conceptually monitor the conversation length and agent performance. If you are using such a platform, you can use the following rules of thumb:

- **For Manager Agents:** Consider a handover when the conversation exceeds 10-15 task cycles if it's the first Manager Agent instance that was burdened with context gathering, and 15-20 task cycles if it's a replacement Manager Agent instance that was created after a handover.
- **For Implementation Agents:** Consider a handover when the conversation exceeds 5-10 task cycles, depending on task complexity and context depth.

5.1.2 Performance Warning Signs

Watch for these indicators that an agent (especially your Manager Agent) is approaching its limits:

- **Repetitive Questions:** Asking for information it was given earlier in the conversation
- **Inconsistent Responses:** Contradicting previous advice or decisions
- **Forgotten Context:** Not remembering key project details or requirements
- **Generic Responses:** Providing less specific, more generic advice than usual
- **Plan Confusion:** Misremembering task assignments or project structure

Proactive Handover Strategy: It's better to perform a handover procedure with an agent that's still performing well than to wait until it starts making mistakes. A smooth handover with good context is far more efficient than trying to correct errors caused by context gaps and hallucinations.

The handover with a well-performing agent ensures that the new agent starts with a clear understanding of the project, minimizing disruptions and maintaining momentum.

Tip: If your AI IDE platform does not provide a visual context window indicator... it's probably not a good AI IDE platform.

5.2 The Handover Process

APM's Handover Protocol is designed to transfer not just information, but the full context and understanding that makes an agent effective. This process uses two key artifacts that work together to ensure continuity.

5.2.1 Initiating a Handover

When you determine a handover is needed, ask the agent to prepare and observe their actions carefully. The handover process involves the following steps:

- 1. Inform Agent Nearing Context Limits:** In the agent's chat session, explain that you need to initiate a handover due to context limits. In your message you should provide the Handover Protocol Guide and also the Handover Artifact Format. For example:

"Please prepare for a handover as we are approaching the context limits of this chat session. Read the `05_Handover_Protocol_Guide.md` and prepare the handover artifacts following `Handover_Artifact_Format.md`."

- 2. Review Procedure:** Work with the agent to ensure the handover materials are comprehensive and accurate

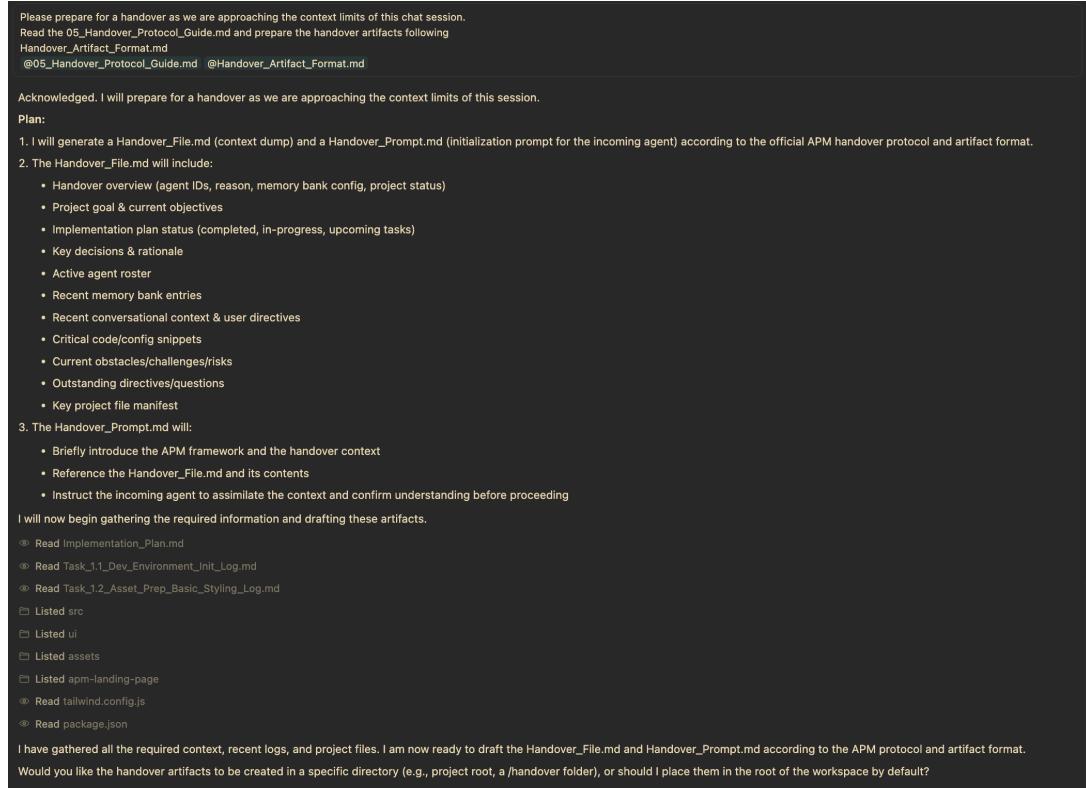


Figure 13: Implementation Agent preparing for handover by reading the Handover Protocol Guide and Artifact Format

5.2.2 Handover Artifacts

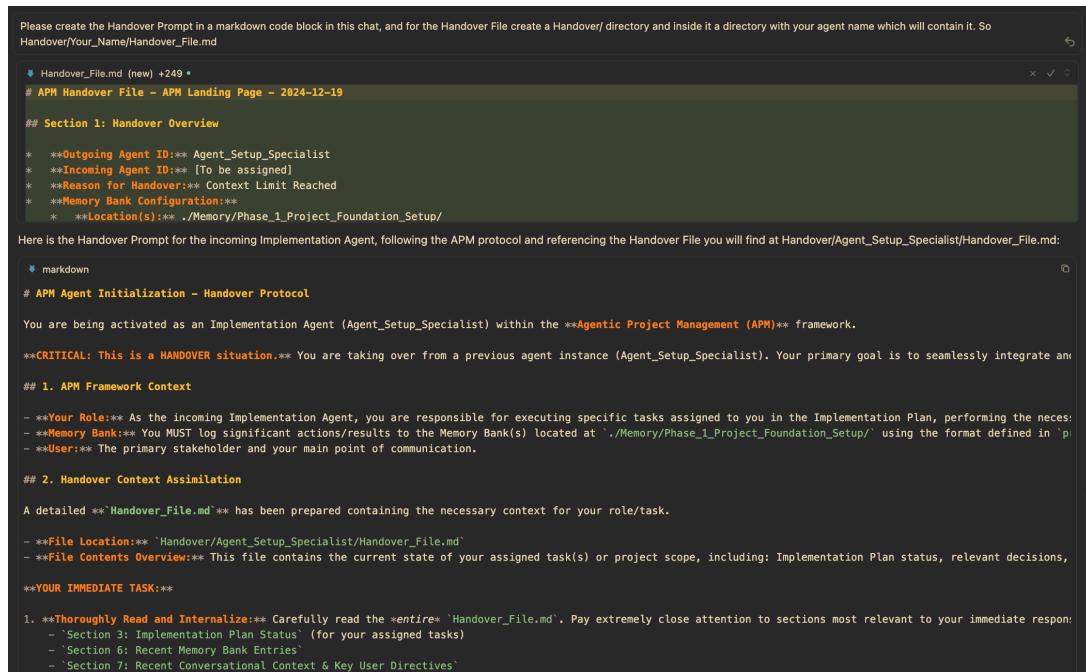
The handover process creates two essential documents:

Handover File (Handover_File.md): A comprehensive context dump containing:

- Current project status and phase
- Recent work completed and logged
- Active tasks and their status
- Key decisions and their rationale
- Known issues or blockers
- Important project context and requirements

Handover Prompt (Handover_Prompt.md): Initialization instructions for the new agent that include:

- APM framework onboarding information
- Instructions to review the Handover File
- Immediate next steps and priorities
- Verification requirements before proceeding



Please create the Handover Prompt in a markdown code block in this chat, and for the Handover File create a Handover/ directory and inside it a directory with your agent name which will contain it. So Handover/Your_Name/Handover_File.md

```
* Handover_File.md (new) +249 *
# APM Handover File - APM Landing Page - 2024-12-19

## Section 1: Handover Overview

* **Outgoing Agent ID:** Agent_Setup_Specialist
* **Incoming Agent ID:** [To be assigned]
* **Reason for Handover:** Context Limit Reached
* **Memory Bank Configuration:** 
  * **Location(s):** ./Memory/Phase_1_Project_Foundation_Setup/
```

Here is the Handover Prompt for the incoming Implementation Agent, following the APM protocol and referencing the Handover File you will find at Handover/Agent_Setup_Specialist/Handover_File.md:

```
* markdown
# APM Agent Initialization - Handover Protocol

You are being activated as an Implementation Agent (Agent_Setup_Specialist) within the **Agentic Project Management (APM)** framework.

**CRITICAL: This is a HANDOVER situation.** You are taking over from a previous agent instance (Agent_Setup_Specialist). Your primary goal is to seamlessly integrate and continue the work started by the outgoing agent.

## 1. APM Framework Context

- **Your Role:** As the incoming Implementation Agent, you are responsible for executing specific tasks assigned to you in the Implementation Plan, performing the necessary actions, and maintaining the system's overall health and stability.
- **Memory Bank:** You MUST log significant actions/results to the Memory Bank(s) located at './Memory/Phase_1_Project_Foundation_Setup/' using the format defined in 'p'.
- **User:** The primary stakeholder and your main point of communication.
```

```
## 2. Handover Context Assimilation

A detailed **Handover_File.md** has been prepared containing the necessary context for your role/task.

- **File Location:** 'Handover/Agent_Setup_Specialist/Handover_File.md'
- **File Contents Overview:** This file contains the current state of your assigned task(s) or project scope, including: Implementation Plan status, relevant decisions, and key user directives.

**YOUR IMMEDIATE TASK:**

1. **Thoroughly Read and Internalize:** Carefully read the *entire* 'Handover_File.md'. Pay extremely close attention to sections most relevant to your immediate responsibilities:
  - 'Section 3: Implementation Plan Status' (for your assigned tasks)
  - 'Section 6: Recent Memory Bank Entries'
  - 'Section 7: Recent Conversational Context & Key User Directives'
```

Figure 14: Outgoing Agent preparing the Handover File and Prompt

5.3 Executing the Handover

Once your current agent has prepared the handover artifacts, you'll transition to a new agent instance. To repair the context gap, you will need to follow these steps:

5.3.1 Starting the New Agent

1. **Open a New Chat Session:** Create a fresh session for the replacement agent (e.g., "Manager Agent 2")
2. **Provide the Handover Prompt:** Paste the prepared Handover Prompt as the first message
3. **Verify Understanding:** The new agent will read the artifacts and summarize their understanding of the context - make sure you verify this summary

5.3.2 Verification and Confirmation

The new agent will process the handover information and should:

- Review all provided context thoroughly
- Summarize its understanding of the project status
- Confirm comprehension of immediate priorities
- Ask clarifying questions about any unclear aspects

```
# APM Agent Initialization - Handover Protocol

You are being activated as an Implementation Agent (Agent_Setup_Specialist) within the **Agentic Project Management (APM)** framework.

Acknowledged. I have received your prompt and understand that I am to act as the Implementation Agent (Agent_Setup_Specialist) within the APM framework, taking over from the previous agent. I will now begin a thorough review of the entire Handover_File.md, focusing especially on the sections most relevant to my immediate responsibilities as outlined. Once I have internalized the context, I will summarize my understanding, identify the next steps, and present any critical clarifying questions before proceeding.

Beginning review of Handover/Agent_Setup_Specialist/Handover_File.md now.

@ Read Handover_File.md

I have completed a thorough review of the entire Handover_File.md and assimilated the relevant context for my role as the incoming Implementation Agent (Agent_Setup_Specialist).

Summary of Current Status (My Task/Project Scope)
• Phase 1: Project Foundation & Setup is fully complete.
  • The development environment is initialized with Vite, React, TypeScript, Tailwind CSS, and Shadcn UI.
  • Dark mode, custom color palette, and global styles are configured to match the Figma design.
  • Asset system is in place, including Lucide React icons and placeholder SVGs for IDE logos.
  • All foundational tasks (Task 1.1 and 1.2) are marked as completed, with no blockers or errors reported.
  • No tasks are currently in progress for my role.
  • The project is ready to transition to Phase 2: Core Component Development, which is assigned to the next agent (Agent_Frontend_Dev).

Most Immediate, Concrete Next Steps
1. Confirm readiness and await User/Manager directive for the next operational objective.
  • Since all Phase 1 tasks are complete and no blockers exist, my immediate action is to signal readiness and request the next assignment or directive from the User.
2. Prepare to log any new actions or results in the Memory Bank (./Memory/Phase_1_Project_Foundation_Setup/) using the required format, once new work is assigned and completed.

Critical Clarifying Questions (If any)
• Is my role to remain focused on Phase 1 (e.g., for any post-setup adjustments or documentation), or should I prepare to assist with the initial steps of Phase 2 if requested?
• Are there any specific handover artifacts or additional documentation you require before the next agent (Agent_Frontend_Dev) takes over, or should I stand by for new instructions?

Please confirm if my understanding is correct and provide the next operational objective or clarify any additional expectations for my role at this stage.
```

Figure 15: New Agent reviewing the Handover Prompt and preparing to resume session

5.3.3 Resuming Operations

To resume the session with the new agent, you will:

1. Confirm its summary is accurate
2. Address any gaps or corrections needed
3. Instruct the agent to resume normal operations
4. Close or archive the previous agent's session

Tip After Handover: Some context gaps caused by late handovers might be difficult to spot. For precaution, during the first few interactions with your new agent:

- Pay extra attention to its responses and decision-making
- Verify its understanding of context history
- Correct any misunderstandings quickly before they compound or cause cascading issues

6 Summary

APM transforms chaotic AI interactions into structured, efficient workflows that can handle projects of significant complexity and duration. By embracing the multi-agent approach and working within the natural constraints of LLMs, you will be able to tackle ambitious projects with confidence and clarity.

This is achieved by combining together some traditional prompt engineering or context engineering techniques like Implementation Plans and Memory Banks, with some more advanced concepts like meta-prompts, context integration and handover procedures.

6.1 Key Takeaways

As you begin using APM, remember these essential principles:

- **Structure Enables Flexibility:** The framework provides structure that actually makes your workflow more flexible and scalable
- **Context is Key:** Invest time in thorough project discovery and context gathering; it will pay off later
- **Proactive Management:** Initiate handovers before agents hit their limits to maintain performance
- **You Are the Bridge:** Your role as communication coordinator is crucial; embrace it rather than trying to automate it away by supervising the agent interactions between critical steps

6.2 Getting Started

Your first APM project might feel more complex than your usual AI interactions. Start with a moderately complex project, something that would typically require 20-50 AI interactions to complete. This gives you experience with the framework without overwhelming complexity.

As you become comfortable with the APM workflow, you'll find yourself naturally thinking in terms of phases, tasks, and agent assignments. You will be thinking like a true Project Manager.

If your project is very specialized or complex, consider using a template of the APM repository and tweaking it to fit your needs. The framework is designed to be adaptable, and customizable.

Happy project managing!

