Module 4 Portfolio Milestone: Project Codebase Design and Outline

Alexander Ricciardi

Colorado State University Global

CSC501: Management for the Computer Science Professional

Dr. Brian Holbert

December 6, 2025

Module 4 Portfolio Milestone: Project Codebase Design and Outline

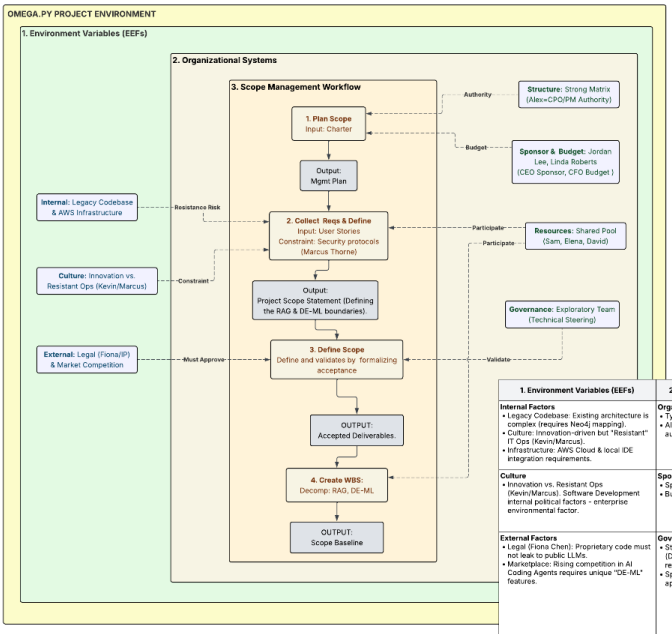
The objective of the Project Codebase Project Management Plan (PMP) is to manage the development and implementation of an intelligent AI agent and RAG (Retrieval-Augmented Generation) system to assist Omega.py software engineers in designing and implementing software applications. Project Codebase's features include visual architecture mapping, project workflow visualization, context descriptions, and integration with IDE (Integrated Development Environment) tools, as well as a custom markup language called DE-ML (DescriptionExtractor Markup Language) integrated within code comments and docstrings for component description extraction. These features' functionality will be powered by Project Codebase AI's own agent to enable AI coding agents and software engineers to understand the constraints, requirements, and stakeholder needs of software projects, as well as their codebase and architecture. This document provides an outline of Project Codebase PMP, including the project scope and timeline estimates for inception, delivery, and maintenance, as well as discussing possible change requests and goal alignments.

Project Scope

Project Scope Statement

The project aligns with Omega.py's values of System for Value Delivery, which state that the value provided by a software is the ultimate indicator of project success or failure. Based on these values, the project's core objective is to develop a sophisticated AI agent and RAG system that will assist Omega.py software engineers and AI coding agents in understanding software codebases and architectures. The goal of the project is to deliver business value by translating employee workflow efficiency into financial savings.

Figure 1
Initial Project Codebase PMP



1. Environment Variables (EEFs)	2. Organizational Systems (Governance)	3. Scope Processes (Workflow)
Internal Factors • Legacy Codebase: Existing architecture is complex (requires Neo4j mapping). • Culture: Innovation-driven but "Resistant" IT Ops (Kevin/Marcus). • Infrastructure: AWS Cloud & local IDE integration requirements.	**Organizational Structure** • Type - Strong Matrix. • Alex Ricciardi acts as both CPO and PM, high authority over resources (Sam, Elena).	**Step 1: Plan Scope** • Input: Project Charter. Action: CEO Authorization (completed), initial budget • Action: Define how the scope will be validated by the "Exploratory Team." • Output: Plan for the Scope Management Plan.
Culture • Innovation vs. Resistant Ops (Kevin/Marcus). Software Development internal political factors - enterprise environmental factor.	**Sponsor & Budget** • Sponsor - Jordan Lee (CEO). • Budget - Linda Roberts	**Step 2: Collect & Define** • Input: Plan for the Scope Management Plan and User Stories from Software Engineers. • Constraint: Security protocols (Marcus Thorne). • Output: Project Scope Statement (Defining the RAG & DE-ML boundaries).
External Factors • Legal (Fiona Chen): Proprietary code must not leak to public LLMs. • Marketplace: Rising competition in AI Coding Agents requires unique "DE-ML" features.	**Governance Framework** • Steering Committee - The "Exploratory Team" (David, Sam, Elena) acts as the technical review board. • Sponsor - Jordan Lee (CEO) provides budget approval. **Resources: Shared Pool** • Leads: Sam (Software), Elena (AI), David (Architect). • Dev Team: Priya, Mark, Sarah (Developers). • Data: Linda Wu (Data Engineer).	**Step 3: Define Scope** • Input: Project Scope Statement (Defining the RAG & DE-ML boundaries) • Key Branches: RAG Agent, DE-ML Parser, Knowledge Graph. • Validates: - Action: Formalizing acceptance (Obtaining Sign-off). - Detail: Reviewing deliverables with the "Exploratory Team" (Elena/Sam) to ensure they meet the acceptance criteria. Output: Accepted Deliverables (Signed-off). **Step 4: Create WBS** • Input: Accepted Deliverables. • Decomposition: RAG, DE-ML, ect. • Create: WBS Dictionary. • Output: Scope Baseline

Note: The diagram illustrates how the project will be managed, and the table gives a detailed breakdown of the work packages and scope components.

Project requirements

Please refer to the table within Figure 1 for a detailed breakdown of the work packages and scope components. The following is the core requirements list:

- Knowledge Graph Database - Implementation of a Neo4j database to map the codebase and architecture of a software application.
- Project Codebase AI Agent - A customized AI backend agent for Project Codebase to assist software engineers and AI coding agents capable of querying Neo4j knowledge graph database, understanding constraints, requirements, and codebase context.
- RAG System - Manage the Neo4j knowledge graph database and RAG Ingestion Pipeline used by the Project Codebase AI Agent.
- Description Extractor Markup Language (DE-ML) – integrated within the RAG for code comment parsing, helping extraction by describing codebase components.
- VS Code Plugin – Project Codebase needs to be deployed as a 'chat' extension within VS Code.
- Programming Language Requirements
    - Python – Required for AI Development, which involves AI Agents and RAG pipelines.
    - TypeScript / JavaScript - Required for IDE Plugin Development.
    - Cypher (Query Language) - Required for creating, maintaining, and querying, Neo4j Knowledge Graph database.
- DE-ML (Proprietary Syntax): The project explicitly mandates the creation of a custom syntax called Description Extractor Markup Language (WBS 3.1) to be embedded within code comments.
- Exclusions - Browser functionality, mobile applications, or standalone desktop applications are currently out of scope.

Scope Baseline and WBS

The scope baseline artifact is primarily gold to control the scope by preventing scope creep and gold plating. The scope baseline includes components, which are the Scope Statement, the Work Breakdown Structure (WBS), and the WBS Dictionary (not provided). The Scope Statement was provided at the beginning of the Project Scope section. Figure 1 provides a decomposition of the scope into work packages, creating the initial WBS component.

The WBS

For Project Codebase WBS initial component breakdown can be listed as follows:

1.0 Plan Project Management:

- 1.1 Finalize Project Charter (completed).
- 1.2 CEO Authorization (completed), initial budget.
- 1.3 Define how the scope will be validated by the "Exploratory Team."
- 1.4 Create Scope Management Plan.

2.0 Infrastructure & Security - Addressing "Resistance Risk":

- 2.1 AWS Cloud Environment Setup.
- 2.2 Neo4j Knowledge Graph Instance Provisioning.
- 2.3 Legal & IP Compliance Audit.

3.0 Core AI Development:

- 3.1 Define DE-ML (Description Extractor Markup Language) Syntax.
- 3.2 Develop RAG Ingestion Pipeline.

4.0 Interface & Client Integration

- 4.1 IDE Plugin Development (VS Code):
  - 4.1.1 Authentication Module (LDAP/SSO).
  - 4.1.2 Context Menu Actions ("Explain Code").
  - 4.1.3 Inline Comments Renderer (Ghost text).
- 4.2 Architecture Visualization Mapper:
  - 4.2.1 Graph Rendering Engine (Frontend).
  - 4.2.2 Interactive Navigation (Click-to-Expand).

5.0 System Integration & Build

- 5.1 RAG Pipeline Integration: Retriever-Generator Link logic.
- 5.2 Description Extractor Integration: Parser-Database Hook.
- 5.3 Security Integration: PII Redaction Layer (Satisfying Security Lead requirements).

Development Methodology and Project Schedule (Timeline)

A hybrid/adaptive model methodology was selected for this project; additionally, the Project Schedule (timeline) integrates this hybrid methodology within the project life cycle. Therefore, the WBS follows the project life cycle phases

Table 1
Hybrid Development Approach

Project Phase	WBS Alignment	Development Approach	Delivery	Key Deliverables
Inception/ Planning	1.0, Initial 2.0	Predictive/ Sequential	Single Document	Establish the baselines.
Core AI Delivery	3.0, 4.0	Adaptive/Iterative	Periodic 2-week Sprints	MVP (Minimum Viable Product) increments.
Integration/ Deployment	5.0	Hybrid/Incremental	Multiple Modules/ Features)	Release tested functionality incrementally (e.g., Auth Module 4.1.1) to users.
Maintenance/ Operations	Post-Project	Sustaining Operations	Continuous/ On-Demand	Feature refinement (Product Backlog Grooming) and continuous defect repair as part of the product life cycle.

Note: The table illustrates the hybrid development approach integrated within the WBS.

Hybrid/Adaptive Model

The hybrid/adaptive model is a hybridization between the classic predictive model (waterfall) and the more modern Agile model. The predictive model (waterfall) was chosen for well-defined tasks and

sequentially driven phases, such as the Plan Project Management phase (WBS 1.0) and the Infrastructure/Security (WBS 2.0) phase for legal and infrastructure compliance constraints. On the other hand, the adaptive model (Agile/iterative) was chosen for core development, such as the AI development (WBS 3.0, 4.0), which involves high technical complexity, unknown, and uncertainty related to the development and implementation of AI systems. The Agile model is well-suited for managing projects with high levels of unknown and uncertainty; its adaptivity approach will help the team to quickly adapt to changes caused by unknown and uncertain factors. Furthermore, its iterative approach, utilizing short sprints, will help break down the technical complexity of this project, making it more manageable. Therefore, as shown in Table 1, this project life cycle follows this hybrid approach.

Project Life Cycle Timeline Estimates - Project Schedule

The project total duration is estimated at 44 Weeks, about 11 months, with a buffer of 2 weeks. Note that the maintenance phase has a life duration of 3 to 5 years. See Table 2 for an illustration of the project life cycle timeline estimates.

Table 2
Project Life Cycle Timeline Estimates

Phase	Activity	Duration	Delivery
Inception WBS 1.0, 2.0	Phase 1: Initiation & Planning	4 Weeks	project charter, stakeholder register, WBS, and environment setup.
Delivery WBS 2.0, 3.0, 4.0	Phase 2: Core Development	32 Weeks	Neo4j setup, AI model training, API development, developing VS Code plugin, and documentation
Integration/ Deployments WBS 5.0	Phase 3: Integration & Deployment	8 Weeks	Deploy VS Code plugin, beta testing with the internal team, and update documentation.
Maintenance/ Handoff	Phase 4: Closure & Handoff	Ongoing: 3-5 Years	Final documentation, ongoing maintenance, user training, transition to ops.

Note: The table illustrates the project life cycle timeline estimates and the phase deliveries.

Change Request Management

To handle major changes, the project management utilizes a formal Change Control Process as defined in Integration Management. The process is as follows.

1. Any stakeholder must submit a Change Request Form describing the changes needed or wanted and their possible business value.
2. Next, the PM (Alex) performs an Impact Analysis, analyzing the potential impact that the requested change can have on the Scope, Schedule, and Cost.
3. The Change Control Board (likely Jordan Lee and the Steering Committee) reviews the request.
4. After a decision is made, the approved changes are integrated within the Project Management Plan; rejected changes are logged.

The Project Plan

Stakeholder Analysis

A Stakeholder Engagement Assessment Matrix was used to assess \stakeholder engagement, see Table 3. The matrix used the PMI-identified levels to categorize the level of engagement. The matrix is a case study that illustrates the stakeholders' current and desired engagement levels

Table 3
Stakeholder Engagement Assessment Matrix

Stakeholder	Unaware	Resistant	Neutral	Supportive	Leading
Fiona Chen (Legal Counsel)	C			D	
Marcus Thorne (Security Lead)		C		D	
Kevin Brooks (Network Admin)		C		D	
Sam Lopes (Lead Software Engineer/Dev.)				C	D
David Kim (Software Architect)				C	D
Priya Patel (Senior Software Developer)				C, D	
Mark O'Connor (Senior Software Engineer)			C	D	
Sarah Jenkins (Junior Software Developer)	C			D	
Linda Wu (Data Engineers)		C		D	
Elena Vance (Lead AI Engineer)				C	D
Jordan Lee (CEO)				C, D	
Linda Roberts (CFO)			C	D	
Alex Ricciardi (CPO, PM)					C, D

C: Current Engagement level
D: Desire Engagement level

Note: The matrix illustrates the current and desired engagement levels of stakeholders.

Stakeholder Engagement Assessment Case Study

The main goal of the Program Manager (PM) is to try to have stakeholders at their desired level. Within the Stakeholder Engagement Assessment Matrix, this translates to having all the Cs and Ds in the same box. Thus, the primary purpose of this Engagement Assessment Matrix case study is not only to assess the stakeholders' current and desired engagement levels but also to formulate targeted strategies that will help stakeholders not currently at their desired level transition to their desired engagement level. Below, the study lists the stakeholders from the currently least engaged to the most engaged:

Currently Unaware

- Fiona Chen is Omega.py's Legal Counsel. She is currently unaware that the project exists, as legal counsel is rarely involved in the early planning and development phases of a software project. However, as the project plans to use external AI models (e.g., Gemini or ChatGPT), which introduces issues with licensing and a liability risk due to possibly leaking clients' sensitive data and Omega.py's sensitive data

to AI providers, her desired engagement level must be Supportive to formulate contracts of use with AI providers that ensure that Omega.py has strict governance over the data shared with the Ai model, and has intellectual property rights over all the outputs generated by the AI model. To transition Fiona to a Supportive level, she needs to be aware of the project RAG system functionality and goals and briefed about the external AI model's current terms of service related to data confidentiality and security.

- Sarah Jenkins is a junior software developer. She is currently unaware that the project exists. As a junior developer, her main focus is not on planning projects or on helping develop or engineer high-level software structures, but on bug fixes and learning the Omega.py coding culture and software development practices. Nonetheless, her desired level of engagement needs to be supportive as she is expected to use Project Codebase extensively for coding tasks. To transition Sarah to the Supportive level, she needs to be aware of the project and train on DE-ML (Description Extractor Markup Language) is being developed, as well as how to test/use the Project Codebase during its various developing phases.

Currently Resistant

- Marcus Thorne is the security lead for Omega.py. He is currently resistant to the project. As a security lead, he is responsible for reducing risks and protecting Omega.py's systems, sensitive information, data, intellectual property, and facilities. In consequence, he sees the introduction of an external system, such as an external AI model having access to client and internal data, as well as network devices, as a security risk that can potentially result in massive "Data Leakage," a security breach of Omega.py systems. His desired engagement level must be Supportive for the project to be viable. To transition Marcus to a Supportive level, he must be actively included in planning, implementing, and reviewing security protocols, safety features, and tools securing the Project Codebase functionality and surrounding infrastructure.

- Kevin Brooks is the Omega.py network administrator. He is currently resistant to the project. As a network administrator, he worries about the network infrastructure's performance, reliability, and security. Introducing a system like Project Codebase that integrates an RAG system, servers for databases, and external API calls to AI model providers will increase network traffic and latency risks, as well as authentication and security issues. His desired engagement level must be Supportive for the project to be effectively and safely integrated within Omega.py infrastructure. To transition Kevin to a Supportive level, he must be involved in the planning and implementation of local or cloud-based Neo4j databases, the API gateway configuration, and Project Codebase authentication protocols.

- Linda Wu is Omega.py data engineer. She is currently resistant to the project. She is responsible for maintaining Omega.py's data databases and the overall data pipeline. She is wondering why, in addition to existing SQL databases, which are a type of relational database, there is a need for Graph Databases, which are also a type of relational database. In other words, she views the implementation of new Neo4j Graph Databases to support Project Codebase alongside existing SQL databases as potentially unnecessary, increasing the complexity and overhead of the data pipeline. Her desired engagement level must be Supportive for the project to be viable. To transition Linda to a Supportive level, she needs to be briefed about why Graph Databases are necessary for the efficient functioning of Project Codebase. For example, by demonstrating how an AI agent can autonomously perform schema mapping, data injection, and query a Graph database within an RAG system. She should also have ownership of the Knowledge Graph template schema that Project Codebase AI needs to follow; this will further her engagement with the project.

Currently Neutral

- Mark O'Connor is an Omega.py senior software engineer. He is currently neutral to resistant to the project. As an experienced developer, he has seen many utility tools come and go. His neutrality on Project Codebase is based on a wait-and-see approach. However, he is skeptical about the value of "visual mapping architecture" using a tool, when manually creating UML diagrams is sufficient in most cases. Additionally, he feels uncomfortable about using AI; however, he is impressed by how much AI coding agents have improved in recent months, making them useful tools in specific cases. His desired engagement level is Supportive, as he needs to be part of the Project Codebase developing team. To transition Mark to a Supportive level, he needs to be further exposed to AI coding agents and to the abilities of AI to understand and visually represent systems. By enrolling him in RAG training/bootcamp through Neo4j and by giving him ownership of a pilot Project Codebase RAG system.

- Linda Roberts is Omega.py Chief Financial Officer (CFO). She is currently neutral on the project. She is aware of the project's existence, but only from a financial perspective; she is only aware of the estimated cost of implementing the project, such as forecasted AI API tokens, cloud hosting/local hosting, and estimated development hours. She is waiting to see (further analysis) if the project yields real value in the long term for the organization. Her desired level is Supportive. Presenting to her a cost-benefit analysis of the project will show how the implementation of an AI agent system can reduce the training time for new hires (like Sarah Jenkins) and decrease the time senior engineers spend on documentation and troubleshooting issues. Showing that implementing such a system is very valuable to the organization because it translates "employee workflow efficiency" into "dollar savings."

Supportive Moving to Leading

- Sam Lopes is Omega.py lead software engineer, David Kim is Omega.py software architect, and Elena Vance is the Omega.py lead AI engineer. They are currently supportive of the project. They were all part of the initial exploratory team for the project. They see the value of the project. However, they have been assigned to lead the project and need to be informed of and empowered to assume their leading role.

Supportive and Lead Already Aligned

- Priya Patel is an Omega.py senior developer. He is currently aligned at a Supportive level. He is an AI enthusiast, seeing the value of the project.

- Jordan Lee is Omega.py CEO. The Project Codebase exploratory team presented him with a Project Codebase proposal. He is very supportive of the project, seeing the value in it. He is currently aligned at a Supportive level.

- Alex Ricciardi is Omega.py Chief Product Officer (CTO) and Program Manager (PM) of Project Codebase. He is the source of the Project Codebase idea and the main cheerleader. He is currently aligned at a Leading level.

Responsibility Assignment

A Responsibility Assignment Matrix (RAM) using the RACI chart is used as an organizational tool for clarifying the team member roles at the work package level. See Table 4 for a High-Level RACI Matrix.

Table 4
High-Level RACI Matrix

WBS Component	Alex Ricciardi - PM/CPO	Elena Vance -Lead AI Eng.	David Kim – Architect	Marcus Thorne - Security	Linda Wu - Data Eng	Sam Lopes - Lead SW Eng.
1.0 Plan Project Management	A	C	C	I	I	C
2.0 Infrastructure & Security	I	C	R	A	C	I
3.0 Core AI Development	I	A, R	C	I	A	C
4.0 Interface & Client Integration	I	C	A	I	I	R
5.0 System Integration & Build	A	R	C	C	C	R
WBS 2.3 Legal/IP Audit	A	I	I	R	I	I
R = Responsible A = Accountable C = Consulted I = Informed						

Note: The matrix illustrates a High-Level RAM using the RACI format to clarify team member roles within WBS components.

The RACI matrix uses dual accountability that can be used to change a team member from resistance into ownership.

Project Budget (Cost Management)

The project budget was created using Bottom-Up Estimating, aggregating costs from the individual work packages defined in the WBS. Note that the total Budget at Completion (BAC) allocates funds across three categories, which are Direct Labor, Infrastructure/Materials, and Risk Reserves. See Table 5 for a preliminary project budget. estimates

Table 5
Preliminary Project Budget Estimates

Cost Category	Item / WBS Reference	Estimate	Justification / Strategy
Direct Labor	Core Development Team	$240,000	3,000 hours @ $80/hr Team: Elena, Sam, David, Priya
Direct Cost (Training)	Resource Acquisition Strategy	$8,500	Mandatory RAG/Neo4j Bootcamp for Mark O'Connor -> Risk Mitigation and DE-ML training for Sarah Jenkins.
Infrastructure	AWS Cloud & Neo4j Licenses	$12,000	WBS 2.0 Environment setup and hosting for Dev/Test.

Variable Costs	AI API Tokens (Gemini/OpenAI)	$15,000	Volatile Cost -> Estimated usage for testing RAG ingestion and inference.
Reserves	Contingency Reserve (10%)	$27,550	Allocated for Known Risks -> token price fluctuation, integration delays
Total	Budget at Completion (BAC)	$303,050	

Note: The table illustrates the preliminary project budget estimates broken down by cost category, item justification, and strategy. It also includes a contingency reserve to mitigate known risks such as price fluctuations and integration delays.

Budget Brief Description

Volatile Cost Management: The Variable Costs for AI API tokens, $15,000, represent a high-volatility estimate. Because Large Language Models (LLMs) cost of use is consumption-based and prone to change. This will be monitored weekly during the Adaptive/Iterative phase (WBS 3.0) to prevent overrun.

Contingency Reserve: A 10% reserve, $27,550, addresses “Known Risks,” such as potential price fluctuations in cloud infrastructure

Brief Summary

This PMP Design and Outline establishes a management framework for the successful development and implementation of Omega.py Project Codebase. By adopting a Hybrid Methodology, the plan balances legal and security compliance with the agility and adaptability required to develop and implement AI systems. By integrating specific Stakeholder Engagement Strategies, such as the “Dual Accountability” in the RACI matrix and funded training in the Budget, the plan addresses resistance from key technical leaders. This approach will facilitate the delivery of a functional and sophisticated AI agent and RAG system that will translate technical efficiency into long-term financial value for Omega.py.