**Project Statement of Work**

**Generated by:** Requirements Gathering Agent v2.1.3

**Project:** === PROJECT README ===

**Category:** core-analysis

**Generated:** 10/06/2025, 02:11:46

──────────────────────────────────────────────────

The **Requirements Gathering Agent** is an AI-powered Node.js/TypeScript tool designed to automate the generation of PMBOK-compliant project management documentation. By leveraging cutting-edge AI technologies such as Azure OpenAI and Google AI, the tool transforms project context into professional-quality documentation, streamlining workflows for project managers and business analysts while ensuring compliance with industry standards.

Organizations worldwide struggle with the inefficiencies of manual project documentation processes, which often lead to delays, inconsistencies, and compliance risks. Automation of these tasks provides tremendous value by reducing administrative overhead, improving accuracy, and aligning stakeholder expectations.

Manual creation of PMBOK-aligned project documentation is error-prone, time-consuming, and lacks standardization. This impacts project success rates and increases operational costs. The **Requirements Gathering Agent** addresses this gap by automating the generation of high-quality documentation, ensuring compliance, and aligning project deliverables with organizational objectives.

The project will deliver a robust CLI tool that integrates with multiple AI providers and incorporates advanced context management techniques to generate PMBOK-compliant documents. The system will be designed for scalability, modularity, and ease of use, ensuring seamless adoption across various industries and project environments.

• Development of a CLI-based tool for generating PMBOK-compliant project documentation.

• Integration with Azure OpenAI, Google AI, GitHub Copilot, and Ollama for AI-powered content generation.

• Comprehensive project analysis functionality to discover and analyze all relevant documentation beyond README.md.

• Support for professional export formats, including Microsoft Word (.docx) and, in the future, PowerPoint (.pptx).

• Validation against PMBOK 7.0 standards for compliance and consistency.

• Enhanced context management system for efficient utilization of AI token limits.

• Development of non-PMBOK-aligned project documentation.

• Customizations for industry-specific project management frameworks outside PMBOK.

• Native desktop or mobile application development (CLI-only).

• Integration with proprietary AI models outside the supported providers (Azure, Google, GitHub, Ollama).

• **Core PMBOK Documents**: Project charter, stakeholder register, scope management plan, risk management plan, etc.

• **Planning Artifacts**: Work Breakdown Structure (WBS), WBS Dictionary, activity lists, milestone lists, etc.

• **Professional Outputs**: Microsoft Word (.docx) and PowerPoint (.pptx) files for enterprise use.

• **Analysis Reports**: Detailed reports on document quality, PMBOK compliance, and validation metrics.

• The tool must generate PMBOK-compliant documents with a quality score of at least 90/100.

• The system must process up to 2M tokens efficiently when using large AI models such as Gemini 1.5 Pro.

• Output formats must align with enterprise standards for professional presentation (Word, PowerPoint).

**Automate Documentation**: Develop a solution that generates PMBOK-compliant project documentation with 100% accuracy within 5 minutes per document.

**Ensure Compliance**: Validate all generated documents against PMBOK 7.0 standards with actionable recommendations for improvement.

**Enhance Usability**: Provide a CLI interface that enables users to easily configure, run, and export documentation.

• Support multiple AI providers for flexibility in deployment.

• Provide modular architecture for future extensibility.

• Enable comprehensive project analysis for richer documentation.

• **Adoption**: Achieve at least 500 weekly downloads within six months of release.

• **Compliance**: Maintain PMBOK validation success rates of 95% or higher for generated documents.

• **Efficiency**: Reduce the time required to generate full project documentation by 80%.

• All core PMBOK documents can be generated and validated successfully.

• Users can export documents in both Markdown and Microsoft Word formats.

• High-value project context is consistently identified and incorporated in generated outputs.

**CLI Tool**: A fully functional Command-Line Interface tool for generating PMBOK-aligned documentation.

**Documentation Suite**: A complete set of PMBOK-compliant project management documents.

**Professional Export Formats**: Enterprise-ready Microsoft Word (.docx) and PowerPoint (.pptx) files.

**Validation Reports**: Detailed compliance and quality assessment reports for all generated documents.

• Documents must follow PMBOK 7.0 terminology and structure.

• Outputs must be formatted correctly for professional use.

• Validation scores must exceed 90/100.

• **Milestone 1**: Completion of core CLI functionality - Month 2.

• **Milestone 2**: Integration with AI providers - Month 4.

• **Milestone 3**: PMBOK validation and compliance module - Month 5.

• **Milestone 4**: Beta release and feedback collection - Month 6.

• **Milestone 5**: Full release and professional export formats - Month 7.

• Availability of Azure OpenAI and other AI providers for integration.

• Access to high-quality project documentation for analysis.

• Adequate testing and feedback from beta users.

• **PMBOK Framework**: Utilize PMBOK principles for planning, execution, monitoring, and controlling.

• **Agile Development**: Incremental development with iterative feedback cycles.

• TypeScript and Node.js-based development for scalability.

• Modular architecture to support future enhancements.

• Integration testing for robust error handling and validation.

• Comprehensive unit and integration testing using Jest.

• Validation against PMBOK standards to ensure compliance.

• Stakeholder feedback collection during beta testing phases.

• Identify risks related to AI provider availability and performance.

• Implement fallback mechanisms for provider failures.

• Monitor and mitigate risks related to schedule delays.

• Stakeholders will provide comprehensive project documentation for analysis.

• AI providers will maintain consistent service availability.

• End-users will have basic familiarity with CLI tools.

• Limited support for AI models outside Azure, Google, GitHub, and Ollama.

• Maximum token limit utilization per AI provider.

• Development team limited to five members.

• Testing resources restricted to beta users and automated scripts.

• Project completion within seven months from initiation.

• **Project Sponsor**: Oversees funding and strategic alignment.

• **Project Manager**: Manages scope, timeline, and resources.

• **Development Team**: Builds and tests the CLI tool and integrations.

• **Quality Assurance Team**: Validates outputs and ensures compliance.

• **Project Manager**:

- Define project scope and objectives.

- Manage stakeholder communication.

- Monitor project progress and risks.

• **Lead Developer**:

- Design and implement the CLI tool.

- Integrate AI providers.

- Ensure modularity and scalability.

• **Quality Analyst**:

- Validate PMBOK compliance.

- Conduct unit and integration tests.

- Generate quality assessment reports.

• All major decisions require approval from the Project Sponsor.

• Day-to-day decisions are managed by the Project Manager.

• Weekly status meetings with stakeholders.

• Bi-weekly sprint reviews with the development team.

• Documentation updates shared via project repository.

This **Project Statement of Work** follows PMBOK best practices and serves as the foundation for project initiation and stakeholder approval.