# Quality Management Plan: Requirements Gathering Agent Project

**1. Introduction**

This Quality Management Plan (QMP) outlines the processes, procedures, and standards for ensuring the quality of the Requirements Gathering Agent (RGA) software project. The plan adheres to PMBOK 7th Edition guidelines and aims to deliver a high-quality, reliable, and user-friendly application that meets stakeholder expectations and complies with PMBOK standards.

**2. Quality Standards**

* **PMBOK 7th Edition:** The RGA project will adhere to the quality management processes and principles defined in the PMBOK 7th Edition. This includes focusing on the performance domains of the project and adhering to best practices for each process group (Initiating, Planning, Executing, Monitoring & Controlling, Closing).
* **ISO 9126:** Software quality characteristics (Functionality, Reliability, Usability, Efficiency, Maintainability, Portability) will be assessed using ISO 9126 as a reference.
* **Coding Standards:** The project will follow established coding standards (e.g., using a linter like ESLint with a pre-defined configuration) to ensure code readability, maintainability, and consistency.
* **Documentation Standards:** All project documentation, including the QMP itself, will follow a consistent style guide to maintain clarity and ease of understanding.

**3. Quality Objectives**

* **Functionality:** The RGA software shall accurately generate PMBOK-compliant documents based on provided input. All features listed in the README shall function as specified.
* **Reliability:** The RGA software shall operate without failure for an extended period under normal operating conditions. Error handling and recovery mechanisms shall be robust.
* **Usability:** The RGA software shall be easy to use and understand, with intuitive command-line options and clear documentation.
* **Efficiency:** The RGA software shall perform its tasks efficiently, minimizing resource consumption and processing time.
* **Maintainability:** The RGA software shall be easy to maintain, update, and extend. The codebase shall be well-structured and documented.
* **Portability:** The RGA software shall be compatible with various operating systems and AI provider APIs.
* **Security:** The RGA software shall protect user data and API keys securely, adhering to best practices for API key management and data handling. Compliance with relevant security standards will be ensured.
* **PMBOK Compliance:** All generated documents must meet the standards and guidelines of the PMBOK 7th Edition.

**4. Quality Roles and Responsibilities**

* **Project Manager:** Overall responsibility for quality management, ensuring adherence to the QMP.
* **Development Team Lead:** Responsible for code quality, adherence to coding standards, and unit testing.
* **QA Engineer:** Responsible for system testing, integration testing, user acceptance testing (UAT), and reporting on quality defects.
* **Technical Writer:** Responsible for creating and maintaining clear and accurate project documentation.

**5. Quality Deliverables and Processes**

* **Code Reviews:** Regular code reviews will be conducted to identify and address potential quality issues early in the development lifecycle.
* **Unit Testing:** Unit tests will be written for each module to ensure individual components function correctly.
* **Integration Testing:** Integration tests will verify the interaction between different modules and components.
* **System Testing:** System tests will validate the overall functionality and performance of the RGA software.
* **User Acceptance Testing (UAT):** UAT will be conducted with representative users to ensure the software meets their needs and expectations.
* **PMBOK Validation:** A dedicated process will be implemented to validate the generated documents against PMBOK 7th Edition standards. This will involve automated checks and manual reviews.
* **Documentation:** Comprehensive documentation, including user guides, API documentation, and technical specifications, will be created and maintained.

**6. Quality Control Activities**

* **Inspections:** Regular inspections of code, documentation, and test results will be performed to identify and correct defects.
* **Testing:** Various testing methods (unit, integration, system, UAT) will be used to verify the quality of the software.
* **Defect Tracking:** A defect tracking system will be used to manage and track identified defects throughout the development lifecycle.
* **Metrics Tracking:** Key quality metrics (e.g., defect density, test coverage) will be monitored and reported.

**7. Quality Assurance Activities**

* **Quality Audits:** Periodic quality audits will be conducted to assess the effectiveness of the quality management system.
* **Process Improvement:** Continuous improvement initiatives will be implemented to enhance the quality management processes.
* **Training:** Training will be provided to the development team on quality management principles and best practices.

**8. Quality Improvement Approaches**

* **Plan-Do-Check-Act (PDCA):** The PDCA cycle will be used to identify, analyze, and resolve quality issues.
* **Root Cause Analysis:** Root cause analysis will be used to identify the underlying causes of defects and prevent recurrence.
* **Lessons Learned:** Lessons learned from previous projects and iterations will be documented and used to improve future quality management processes.

**9. Quality Tools and Techniques**

* **Defect Tracking System (e.g., Jira):** For managing and tracking defects.
* **Version Control System (e.g., Git):** For managing code changes and ensuring traceability.
* **Test Management Tool (e.g., TestRail):** For planning, executing, and managing tests.
* **Static Code Analysis Tools (e.g., SonarQube, ESLint):** For identifying potential code defects.
* **Code Review Tools (e.g., GitHub, GitLab):** For facilitating code reviews.

**10. Quality Metrics and Acceptance Criteria**

* **Defect Density:** The number of defects per 1000 lines of code will be tracked and kept below a predefined threshold (e.g., < 2).
* **Test Coverage:** Code coverage will be measured and maintained above a specified threshold (e.g., > 80%).
* **PMBOK Compliance Score:** A score will be assigned to each generated document based on its adherence to PMBOK 7th Edition standards (e.g., > 95%).
* **User Satisfaction:** User satisfaction will be measured through UAT feedback and surveys. A minimum satisfaction score (e.g., > 80%) will be required for acceptance.

**11. Quality Documentation Requirements**

* **Quality Management Plan:** This document.
* **Test Plan:** A detailed plan outlining the testing strategy and procedures.
* **Test Cases:** Specific test cases for each test level (unit, integration, system, UAT).
* **Defect Reports:** Reports documenting identified defects and their resolution.
* **Quality Metrics Reports:** Reports summarizing key quality metrics.

**12. Verification and Validation Approach**

* **Verification:** Ensuring that the software is built according to the specifications. This will involve code reviews, inspections, and