## Acceptance Criteria for Requirements Gathering Agent

These acceptance criteria are structured using the Given-When-Then format and address both functional and non-functional requirements. They are categorized by major features for clarity.

**I. Core Functionality: PMBOK Document Generation**

**A. Project Charter Generation:**

* **Given** a valid project README.md file containing project information (title, goals, stakeholders, etc.)
* **When** the requirements-gathering-agent command is executed
* **Then** a valid PMBOK-compliant Project Charter document (e.g., in Markdown or .docx format, depending on the --format flag) is generated in the specified output directory. The document must include all required elements as defined by PMBOK 7.0, including project objectives, stakeholders, assumptions, constraints, and high-level risks.
* **Given** a README.md file with missing or incomplete information.
* **When** the requirements-gathering-agent command is executed.
* **Then** the tool generates a Project Charter document with placeholders or warnings indicating missing information, and the generation process completes without crashing. A clear report detailing the missing information is provided.
* **Given** a large README.md file exceeding 100KB.
* **When** the requirements-gathering-agent command is executed.
* **Then** the Project Charter is generated within a reasonable timeframe (e.g., under 60 seconds).

**B. Other PMBOK Document Generation:** (Repeat A for each document type: Stakeholder Register, Scope Management Plan, etc.)

* **Given** a valid project README.md file and related project documentation (e.g., requirements documents, architecture diagrams).
* **When** the requirements-gathering-agent command is executed with the appropriate flags (e.g., --generate-stakeholder).
* **Then** a valid PMBOK-compliant [Document Type] document is generated, including all required elements as defined by PMBOK 7.0. The document should accurately reflect the information extracted from the source files.
* **Given** inconsistent or contradictory information across multiple source files.
* **When** the requirements-gathering-agent command is executed.
* **Then** the tool identifies and reports the inconsistencies, providing suggestions for resolution or generating a document that flags the inconsistencies. The generation process does not crash.

**C. Multiple Export Formats:**

* **Given** a valid project README.md file.
* **When** the requirements-gathering-agent command is executed with different --format flags (markdown, docx, json, yaml).
* **Then** the tool successfully generates the PMBOK documents in the specified format without errors. The generated files are valid according to the respective format specifications (e.g., well-formed JSON, valid .docx file). For .docx format, the document should be properly formatted and contain appropriate metadata.
* **Given** an unsupported format is specified using --format.
* **When** the requirements-gathering-agent command is executed.
* **Then** the tool displays a clear error message indicating the unsupported format and lists the supported formats.

**II. Enhanced Project Analysis**

**A. Comprehensive Source Discovery:**

* **Given** a project directory with various documentation files in standard locations (README.md, requirements/, docs/, etc.) and non-standard locations.
* **When** the requirements-gathering-agent command is executed.
* **Then** the tool successfully discovers and analyzes all relevant Markdown files within the project directory, including those in non-standard locations. A report indicating the discovered files and their relevance scores is generated.
* **Given** a project with deeply nested directories.
* **When** the requirements-gathering-agent command is executed.
* **Then** the tool successfully discovers and analyzes relevant files up to the configured recursion depth, without exceeding a reasonable processing time.

**B. Smart Relevance Scoring:**

* **Given** a set of project documentation files with varying relevance to PMBOK concepts.
* **When** the requirements-gathering-agent command is executed.
* **Then** each file is assigned a relevance score (0-100) that accurately reflects its relevance to the project management aspects. Files highly relevant to PMBOK concepts receive higher scores. The scoring algorithm should be documented and auditable.

**C. Automatic Categorization:**

* **Given** a diverse set of project documentation files.
* **When** the requirements-gathering-agent command is executed.
* **Then** the tool correctly categorizes each discovered file into predefined categories (Primary, Planning, Development, Documentation, Other) based on content and location. The categorization should be consistent and accurate.

**III. Enhanced Context Manager**

**A. Context Utilization:**

* **Given** a large language model (e.g., Gemini 1.5 Pro) with a 2M token limit and a project with extensive documentation.
* **When** the requirements-gathering-agent command is executed.
* **Then** the Enhanced Context Manager utilizes a significant portion (at least 50%) of the available tokens for context, maximizing the information provided to the LLM. A report showing context utilization is generated.
* **Given** a smaller language model (e.g., GPT-4) with a limited token context window.
* **When** the requirements-gathering-agent command is executed.
* **Then** the Enhanced Context Manager intelligently manages the context, prioritizing the most relevant information and avoiding token overflow, while still providing sufficient context for accurate document generation.

**B. 3-Phase Context Strategy:**

* **Given** various AI models with different token limits.
* **When** the requirements-gathering-agent command is executed.
* **Then** the Enhanced Context Manager automatically selects and applies the appropriate 3-phase context strategy (Core Context, Ultra-Large Model Support, Large Model Supplementary) based on the detected model capabilities. The selection process should be documented and transparent.

**IV. PMBOK 7.0 Compliance Validation**

* **Given** a set of generated PMBOK documents.
* **When** the requirements-gathering-agent command is executed with the --validate-pmbok flag.
* **Then** the tool performs a comprehensive validation against PMBOK 7.0 standards, identifying any missing or incomplete elements. A detailed report summarizing the compliance status and providing actionable recommendations for improvement is generated.

**V. Error Handling and Robustness**

* **Given** various error conditions (e.g., network issues, API rate limits, invalid input data).
* **When** the requirements-gathering-agent command is executed.
* **Then** the tool handles the errors gracefully, providing informative error messages and attempting to recover where possible. The tool should not crash unexpectedly. Retry logic (when enabled) should be implemented correctly.

**VI. Performance**

* **Given** a typical project with a moderate amount of documentation.
* **When** the requirements-gathering-agent command is executed.
* **Then** the entire process (analysis, generation, validation) completes within a reasonable timeframe (e.g., under 5 minutes). The performance should be documented and monitored.

These acceptance criteria provide a comprehensive framework for testing and validating the Requirements Gathering Agent. Specific performance thresholds and time limits may need further refinement based on testing and resource constraints. Additional criteria may be needed to cover specific aspects of the user interface and user experience.