Acceptance Criteria

> **Source File:** generated-documents\technical-analysis\acceptance-criteria.md
> **Generated:** 16/07/2025 at 14:00:01
> **Generated by:** Requirements Gathering Agent - PDF Converter

Acceptance Criteria

Generated by adpa-enterprise-framework-automation v3.2.0
Category: technical-analysis
Generated: 2025-07-15T18:09:05.129Z
Description: Comprehensive acceptance criteria and validation methods

---

Acceptancecriteria

Project: ADPA - Advanced Document Processing & Automation Framework
Version: 3.2.0
Document Date: 2025-07-08
Prepared by: Requirements Documentation Consultant

---

1. General Product Acceptance

1.1 **Modularity and Standards Compliance**

- The framework must support modular integration of document processing components for business analysis, project management, and data management.
- All generated documentation must comply with at least one of: BABOK v3, PMBOK 7th Edition, or DMBOK 2.0 standards.
- Multi-framework integration must enable unified reporting and cross-referencing across BABOK, PMBOK, and DMBOK artifacts.

1.2 **Technology and Platform Requirements**

- The system must be built using Node.js (>=18.0.0) and TypeScript (>=5.7.2).
- CLI tools, REST API, and admin web interface must be available and installable via npm.
- The system must provide detailed installation and configuration documentation for each interface (CLI, API, Admin Web).

---

2. Core Functional Acceptance

2.1 **Document Generation**

- The system must generate professional, standards-compliant documents for:
  - BABOK v3 (requirements elicitation, stakeholder analysis, planning, validation, enterprise analysis)
  - PMBOK 7th Edition (project charter, scope, risk, resource, schedule, and cost management)
  - DMBOK 2.0 (data governance, architecture, quality, security, privacy) — where marked production ready or in progress.
- Document generation must support multiple output formats: Markdown, PDF, JSON.
- Document templates must be configurable and extensible by users.

2.2 **AI-Powered Automation**

- Users must be able to select and configure AI providers from: OpenAI, Google AI, GitHub Copilot, Ollama, Azure OpenAI.

- The system must support automatic failover between AI providers.
- Intelligent context management must be implemented, ensuring relevant context is injected for all document generation tasks.

2.3 **Automated Workflows**

- End-to-end document generation pipelines must be available, automating multi-step processes (input analysis, template selection, AI processing, output formatting).
- Batch processing support for CLI and API must be implemented.

---

3. Interface Acceptance

3.1 **CLI Interface**

- A command-line tool must be provided, supporting:
  - Document generation (`adpa generate`)
  - Integration setup (Confluence, SharePoint, Adobe)
  - Provider selection and configuration
- CLI help and usage documentation must be accessible via `--help` .

3.2 **REST API**

- API must conform to OpenAPI 3.0 specification, auto-generated via TypeSpec.
- API endpoints must include:
  - `/api/v1/generate` – Document generation
  - `/api/v1/templates` – Template management (CRUD)
  - `/api/v1/confluence/publish` – Publish to Confluence
  - `/api/v1/sharepoint/upload` – Upload to SharePoint
  - `/api/v1/frameworks` – Supported frameworks listing
- API must include health, readiness, and metrics endpoints.

3.3 **Admin Web Interface**

- Next.js-based admin portal must allow for management of users, templates, documents, and integrations.

- Portal must be accessible at the configured port (default: 3001) after installation.

---

4. Integration Acceptance

4.1 **Confluence Integration**

- System must support OAuth2 authentication with Atlassian Confluence.
- Users must be able to publish generated documents directly to Confluence spaces and pages.
- Document metadata and versioning must be preserved.

4.2 **SharePoint Integration**

- System must support OAuth2/Azure AD authentication for SharePoint Online.
- Users must be able to upload documents to specified SharePoint folders/libraries.
- Metadata tagging and version control must be supported.

4.3 **Adobe Document Services**

- System must support professional PDF generation using Adobe APIs.
- Where Adobe Creative Suite is enabled, advanced layout (InDesign), visualization (Illustrator), and image enhancement (Photoshop) must be available per the phase 2 architecture.
- OAuth2 authentication and credential management for Adobe APIs must be implemented.

---

5. Compliance, Security, and Enterprise Readiness

5.1 **Authentication and Authorization**

- All interfaces (CLI, API, Admin) must support secure authentication methods (API key, JWT, OAuth2).
- Role-based access control must be enforced for multi-user/admin scenarios.

5.2 **Regulatory Compliance**

- The system must demonstrate controls supporting regulatory requirements: GDPR, SOX, PCI DSS, Basel III, MiFID II, FINRA, HIPAA, FedRAMP, where applicable.
- Audit trails must log user actions (creation, modification, publishing, deletion).

5.3 **Security Best Practices**

- Security headers (Helmet), CORS, rate limiting, and input validation must be enabled by default.
- Sensitive configuration (API keys, secrets) must never be exposed in logs or error messages.

---

6. Performance, Scalability, and Reliability

6.1 **Scalability**

- The system must support deployment in horizontally scalable environments (microservices, Docker/Kubernetes).
- Load balancing and Redis-based caching must be available for high-throughput scenarios.

6.2 **Reliability**

- Health and readiness endpoints must report accurate system status.
- AI provider failover must be automatic and transparent to the user.
- All critical operations (document generation, publishing, uploads) must include proper error handling and retry logic.

6.3 **Testing and Quality**

- Comprehensive unit, integration, and performance test suites must be provided (Jest-based).
- Test coverage reports must be available.
- All major features (document generation, integrations, API endpoints) must pass end-to-end tests.

---

7. Usability and Documentation

7.1 **User Guidance**

- Quick start guides and detailed usage examples must be included for CLI, API, and admin interface.
- Interactive menus and prompts (e.g., for AI provider selection) must be intuitive, with validation and error feedback.

7.2 **Developer Documentation**

- API documentation (Swagger UI/Redoc) must be accessible from the running API server.
- Codebase must include clear inline comments and architectural overviews.

7.3 **Support & Community**

- Links to GitHub Issues, Discussions, and enterprise support must be visible in documentation.
- Contribution guidelines and code standards must be published and enforced.

---

8. Roadmap and Extensibility

8.1 **Extensible Architecture**

- The framework must allow for the addition of new document types, templates, AI providers, and integrations with minimal code changes.

8.2 **Planned Features**

- Progress on DMBOK 2.0 support, Docker/Kubernetes deployment, analytics dashboard, SSO integration, workflow automation, real-time collaboration, and mobile support must be tracked per roadmap.

---

9. Acceptance Verification

- All acceptance criteria must be traceable to user stories, requirements, and/or referenced industry standards.
- Demonstrable evidence (e.g., successful test results, generated documentation samples, UI walkthroughs) must be provided for each acceptance item.
- Stakeholder sign-off is required prior to production release.

End of Acceptance Criteria