Acceptance Criteria

> **Source File:** generated-documents\technical-analysis\acceptance-criteria.md
> **Generated:** 15/07/2025 at 11:40:14
> **Generated by:** Requirements Gathering Agent - PDF Converter

Acceptance Criteria

Generated by adpa-enterprise-framework-automation v3.2.0
Category: technical-analysis
Generated: 2025-07-14T21:25:50.588Z
Description: Comprehensive acceptance criteria and validation methods

---

Acceptancecriteria

Project: ADPA - Advanced Document Processing & Automation Framework

---

1. General Framework Requirements

1.1 Installation & Deployment

- **1.1.1** The framework SHALL be installable via NPM as a global CLI tool (`npm install -g adpa-enterprise-framework-automation`).

- **1.1.2** The source code SHALL be cloneable, buildable, and runnable from GitHub following the documented steps.
- **1.1.3** Docker support SHALL be provided (once released), enabling containerized deployments.
- **1.1.4** The system SHALL provide clear environment configuration templates (`.env.example`), supporting safe and repeatable setups.

1.2 Interface Availability

- **1.2.1** The solution SHALL provide:
  - A CLI interface for all core functions.
  - A REST API, described by TypeSpec/OpenAPI and accessible via Swagger UI.
  - An admin web interface accessible at `http://localhost:3001` after setup.
- **1.2.2** The REST API SHALL run on a configurable port (default: 3000/3001) and expose interactive documentation.

---

2. Standards Compliance & Framework Support

2.1 Business Analysis (BABOK v3)

- **2.1.1** The system SHALL support generation of:
  - Requirements elicitation and analysis documents.
  - Stakeholder analysis and management reports.
  - Business analysis planning artifacts.
  - Solution assessment and validation outputs.
  - Enterprise analysis documentation.
- **2.1.2** BABOK v3 templates SHALL be available and customizable.

2.2 Project Management (PMBOK 7th Edition)

- **2.2.1** The framework SHALL generate PMBOK-compliant documents:

    - Project charters, scope statements, and management plans.
    - Stakeholder management, risk, quality, resource, schedule, and cost documents.
- **2.2.2** PMBOK templates SHALL be accessible via CLI and API.

2.3 Data Management (DMBOK 2.0)

- **2.3.1** The system SHALL provide (at minimum, in-progress):
    - Data governance, architecture, quality, MDM, and security templates.
    - The ability to extend and update as the DMBOK module matures.

2.4 Cross-Framework & Unified Reporting

- **2.4.1** The solution SHALL allow cross-referencing between standards for unified, multi-framework reports.

---

3. AI-Powered Document Generation

3.1 AI Provider Integration

- **3.1.1** The framework SHALL support the following providers:
    - OpenAI (GPT-3.5, GPT-4), Google AI (Gemini), GitHub Copilot, Ollama, Azure OpenAI.
- **3.1.2** Users SHALL be able to configure and switch providers via environment variables or interactive menus.
- **3.1.3** Automatic provider failover SHALL be implemented.
- **3.1.4** The system SHALL validate provider credentials during configuration.

3.2 Context Management

- **3.2.1** The platform SHALL employ intelligent context injection for improved document relevancy.

- **3.2.2** It SHALL manage context per project, user, or session.

3.3 Document Generation

- **3.3.1** Users SHALL generate documents by:
  - CLI: with key/category/format/output options.
  - API: via endpoints (e.g., `POST /api/v1/generate`).
- **3.3.2** Output formats SHALL include Markdown, PDF, JSON, and other industry-standard types.
- **3.3.3** Template-driven generation SHALL allow for custom branding, layout, and compliance with target standards.

3.4 Workflow Automation

- **3.4.1** End-to-end document pipelines SHALL be automated for batch and complex generation scenarios.

---

4. Enterprise Integration

4.1 Confluence Integration

- **4.1.1** The system SHALL support OAuth2 authentication and publishing of generated documents to Atlassian Confluence.
- **4.1.2** CLI and API commands for publishing SHALL be provided (e.g., `adpa confluence publish`).

4.2 SharePoint Integration

- **4.2.1** The platform SHALL enable secure, authenticated publishing to Microsoft SharePoint, including:
  - OAuth2/Graph API-based authentication.
  - Folder structure and metadata management.
  - Version control support.
  - Batch document publishing.

4.3 Adobe Document Services

- **4.3.1** The framework SHALL provide PDF and professional document generation using Adobe APIs.
- **4.3.2** Future phases SHALL support InDesign, Illustrator, and Photoshop integration as per project roadmap.

4.4 Version Control

- **4.4.1** The CLI SHALL support VCS operations, including commit and push (e.g., GitHub, GitLab).

---

5. Security & Compliance

5.1 Security

- **5.1.1** All APIs and integrations SHALL support production-grade authentication (API key, JWT, OAuth2).
- **5.1.2** Security headers, rate limiting, CORS, and input validation SHALL be active by default.
- **5.1.3** Sensitive configuration data SHALL be managed via environment files and not committed to source control.

5.2 Regulatory & Industry Compliance

- **5.2.1** The framework SHALL enable compliance with:
  - Financial: Basel III, MiFID II, FINRA, CFTC, FCA, BaFin.
  - Security: GDPR, SOX, PCI DSS, ISO 27001, ISO 9001.
  - Industry: HIPAA, FedRAMP (where applicable).
- **5.2.2** Documentation and audit logs SHALL be available for compliance verification.

---

6. Scalability, Performance, and Reliability

6.1 Scalability

- **6.1.1** The architecture SHALL support horizontal scaling (microservices, stateless design).
- **6.1.2** Caching (e.g., Redis) SHALL be supported for performance-critical scenarios.

6.2 Performance

- **6.2.1** The system SHALL provide monitoring endpoints and health checks (e.g., `/api/v1/health`).
- **6.2.2** Performance testing scripts SHALL be available and pass predefined benchmarks.

6.3 Reliability

- **6.3.1** The system SHALL handle AI provider failures gracefully and trigger fallback mechanisms.
- **6.3.2** All critical operations SHALL be logged, and error handling SHALL be robust and user-friendly.

---

7. Usability & User Experience

7.1 Documentation & Onboarding

- **7.1.1** Complete user guides, API documentation, and CLI help SHALL be provided.
- **7.1.2** Quick start instructions SHALL enable new users to generate their first document within 10 minutes.

7.2 Admin Interface

- **7.2.1** A web-based admin interface SHALL provide management of templates, jobs, and integrations.

7.3 Interactive Features

- **7.3.1** CLI and admin UI SHALL provide interactive menus for actions such as AI provider selection and template creation.
- **7.3.2** Errors and validation messages SHALL be clear and actionable.

---

8. Testing & Quality Assurance

8.1 Automated Testing

- **8.1.1** The project SHALL include comprehensive unit, integration, and performance tests (e.g., via Jest).
- **8.1.2** All supported AI providers and integrations SHALL have passing test coverage.
- **8.1.3** Every release SHALL pass the full test suite before deployment.

8.2 Template & Document Validation

- **8.2.1** Document templates SHALL be validated for structure and required fields before use.
- **8.2.2** Generated documents SHALL be validated for compliance with selected standards.

---

9. Extensibility & Maintainability

- **9.1.1** The codebase SHALL follow strict TypeScript standards and style guides (Airbnb, Prettier).
- **9.1.2** New standards, providers, and integrations SHALL be easily pluggable with minimal code changes.
- **9.1.3** Configuration SHALL be managed externally, supporting multi-tenant and multi-environment deployments.

---

10. Roadmap Alignment & Future-Proofing

- **10.1.1** Features marked as “in progress” (e.g., DMBOK 2.0, Docker, Kubernetes, mobile support) SHALL have stubs or feature flags for forward compatibility.
- **10.1.2** The framework SHALL track and document roadmap progress, with clear separation of stable and experimental modules.

---

11. Support, Community, & Contribution

- **11.1.1** The project SHALL provide:
  - Public issue tracking.
  - Discussion forums.
  - Contribution guidelines.
  - Contact information for enterprise support.

---

12. Licensing & Acknowledgment

- **12.1.1** The solution SHALL be licensed under MIT, with all dependencies and contributions adhering to open source best practices.
- **12.1.2** Proper acknowledgment of standards bodies, AI providers, and community contributors SHALL be maintained in all major documentation.

End of Acceptancecriteria Document
