Tech Acceptance Criteria

> **Source File:** generated-documents\quality-assurance\tech-acceptance-criteria.md
> **Generated:** 16/07/2025 at 13:58:44
> **Generated by:** Requirements Gathering Agent - PDF Converter

Technical Acceptance Criteria

Generated by adpa-enterprise-framework-automation v3.2.0
Category: quality-assurance
Generated: 2025-07-14T21:05:28.441Z
Description: Technical acceptance criteria and validation requirements

---

Certainly. Below are comprehensive, actionable Technical Acceptance Criteria for the "ADPA - Advanced Document Processing & Automation Framework," aligning with your context and covering all required categories. Each criterion is structured for clarity, verification, and implementation.

Technical Acceptance Criteria – ADPA Enterprise Automation Framework

---

1. Functional Technical Criteria

1.1 API Functionality

- **Acceptance Criteria:**
    1. All REST API endpoints (e.g., `/api/v1/generate`, `/api/v1/templates`, `/api/v1/confluence/publish`) must conform to the OpenAPI 3.0 specification generated via TypeSpec.
    2. API endpoints must validate request payloads using express-validator or Joi/Zod schemas; invalid payloads must return HTTP 400 with error details.
    3. All endpoints must handle errors gracefully, returning standardized error objects with HTTP status codes (400, 401, 403, 404, 500).
    4. API responses must be JSON-formatted and include version, status, and timestamp metadata.
    5. API documentation (`/api-docs`) must automatically sync with implemented endpoints.
- **Test Scenarios:**
    - Submit valid and invalid payloads to each endpoint; verify correct status codes and error responses.
    - Intentionally trigger server errors to check HTTP 500 responses.
    - Review `/api-docs` and compare with OpenAPI schema.
- **Success Metrics:**
    - 100% endpoint coverage in OpenAPI documentation.
    - 100% schema validation coverage.

    - API health and readiness endpoints return status within 500ms.
- **Validation Method:** Automated API integration tests (Jest + Supertest), manual Swagger UI inspection.
- **Failure Conditions:** Undocumented endpoints, missing/invalid schema validation, inconsistent error formats.
- **Acceptance Threshold:** 100% of endpoints pass functional and contract tests; no uncaught exceptions.

1.2 Data Processing

- **Acceptance Criteria:**
    1. All document input data must be validated against JSON schemas.
    2. Template data transformations must produce output conforming to selected framework (BABOK, PMBOK, DMBOK).
    3. Generated documents must be stored and retrievable via the API with unique job/document IDs.
    4. Data retrieval operations must support pagination and filtering.
    5. Data integrity must be maintained across all storage and retrieval operations.
- **Test Scenarios:**
    - Attempt document generation with invalid, incomplete, and valid data.
    - Retrieve generated documents using various filters/pagination.
    - Simulate concurrent generation and retrieval requests.
- **Success Metrics:**
    - 0% data loss or corruption in stress tests.
    - 100% schema conformance for all stored data.
- **Validation Method:** Automated unit/integration tests; data mutation/fuzzing tests.
- **Failure Conditions:** Data corruption, retrieval errors, schema violations.
- **Acceptance Threshold:** 99.99% data integrity in load and concurrency scenarios.

1.3 Business Logic

- **Acceptance Criteria:**
    1. All business rules (e.g., standards compliance, workflow automation) must be implemented as per BABOK, PMBOK, and DMBOK guidelines.
    2. Calculations (e.g., project risk, resource allocation) must be accurate to within 0.1% of reference implementation.
    3. AI provider selection follows configured priority and supports failover.
- **Test Scenarios:**
    - Compare generated documents with reference outputs for the same input.
    - Simulate provider outages to verify failover logic.
- **Success Metrics:**
    - 100% rule coverage in test suites.
    - Calculation results match reference within acceptance margin.
- **Validation Method:** Automated business logic unit tests; output diffing; failover integration tests.
- **Failure Conditions:** Incorrect calculations, unhandled rule violations, AI provider failover failures.

- **Acceptance Threshold:** 100% of rules covered; 0 failed business logic tests.

1.4 Integration Points

- **Acceptance Criteria:**

  1. All external integrations (Adobe, Confluence, SharePoint) must authenticate securely (OAuth2, API Key, or SAML as applicable).
  2. Data exchanged with external services must conform to their API specifications.
  3. Error handling for external APIs must be robust (timeouts, retries, fallback).

- **Test Scenarios:**

  - Publish documents to Confluence/SharePoint and verify presence, metadata, and version control.
  - Simulate authentication failures and network timeouts.

- **Success Metrics:**

  - 100% success in publishing and retrieving test documents from integrations.
  - <2% failure rate in high-volume integration tests.

- **Validation Method:** Automated end-to-end integration tests; manual verification in target platforms.

- **Failure Conditions:** Authentication errors, data loss, protocol mismatches.

- **Acceptance Threshold:** 99% reliability for all integration operations.

1.5 User Interface

- **Acceptance Criteria:**

  1. The Next.js admin interface must render correctly in latest Chrome, Firefox, Edge, and Safari.
  2. All UI elements must be responsive down to 375px width.
  3. Accessibility: All UI components must meet WCAG 2.1 AA standards (color contrast, keyboard navigation, aria-labels).

- **Test Scenarios:**

  - Render admin interface on different browsers/devices.
  - Run Lighthouse accessibility audits.

- **Success Metrics:**

  - Lighthouse accessibility score ≥90.
  - No critical UI rendering bugs on supported browsers/devices.

- **Validation Method:** Automated UI tests (Jest/React Testing Library), manual browser/device testing.

- **Failure Conditions:** Broken layouts, inaccessible controls, browser-specific errors.

- **Acceptance Threshold:** 100% pass rate on supported browsers and accessibility audits.

---

2. Performance Acceptance Criteria

2.1 Response Time

- **Acceptance Criteria:**

    - API endpoints must respond within 500ms P95 for read operations and 2s P95 for document generation under normal load (≤100 concurrent users).
- **Test Scenarios:**
    - Load test with simulated concurrent API requests.
- **Success Metrics:**
    - ≥95% requests meet SLA.
- **Validation Method:**
    - Automated load testing (e.g., Artillery, k6).
- **Failure Conditions:**
    - > *5% requests exceed response time thresholds.*
- **Acceptance Threshold:**
    - 95% requests meet defined response times in load tests.

2.2 Throughput

- **Acceptance Criteria:**
    - System must process at least 50 document generation jobs/minute with ≤5s average latency.
- **Test Scenarios:**
    - Bulk job submission and monitoring.
- **Success Metrics:**
    - ≥50 jobs/minute sustained throughput.
- **Validation Method:**
    - Performance tests with monitoring/logging.
- **Failure Conditions:**
    - Throughput drops below defined threshold.
- **Acceptance Threshold:**
    - 100% throughput targets met under realistic workloads.

2.3 Resource Utilization

- **Acceptance Criteria:**
    - CPU usage must not exceed 75% on reference hardware (4 vCPU/8GB RAM) under normal load.
    - Memory leaks must not occur in 24-hour soak tests.
- **Test Scenarios:**
    - Monitor system during sustained operation.

- **Success Metrics:**
    - CPU ≤75%, memory usage stable.
- **Validation Method:**
    - Observability dashboards (Prometheus/Grafana), soak testing.
- **Failure Conditions:**
    - Resource spikes, memory leaks, process crashes.
- **Acceptance Threshold:**
    - No resource limit violations in monitored period.

2.4 Scalability & Load Handling

- **Acceptance Criteria:**
    - System must scale horizontally to support 1000 concurrent users with <10% performance degradation.
    - Load balancer distributes requests evenly; API remains available during scale events.
- **Test Scenarios:**
    - Simulate user ramp-up and horizontal scaling.
- **Success Metrics:**
    - <10% degradation in key metrics; zero downtime during scale.
- **Validation Method:**
    - Distributed load tests, cloud scaling simulations.
- **Failure Conditions:**
    - Downtime, unbalanced load, excessive degradation.
- **Acceptance Threshold:**
    - Meets scaling and load requirements in non-production and pre-release environments.

---

3. Security Acceptance Criteria

3.1 Authentication

- **Acceptance Criteria:**
    - API/CLI and admin UI must require authenticated access (JWT, OAuth2, or SAML).
    - Session tokens must expire in ≤60 minutes and support secure refresh.
- **Test Scenarios:**
    - Attempt unauthorized access; token expiry and refresh tests.
- **Success Metrics:**
    - 100% of endpoints enforce authentication.
- **Validation Method:**

    - Automated security/API tests.
- **Failure Conditions:**
    - Unauthorized access, token reuse after expiry.
- **Acceptance Threshold:**
    - No unauthorized access permitted.

3.2 Authorization

- **Acceptance Criteria:**
    - Role-based access control (RBAC) implemented for all sensitive operations (document management, integrations, admin).
    - Permission checks enforced at API and UI levels.
- **Test Scenarios:**
    - Test access with various user roles; permission escalation attempts.
- **Success Metrics:**
    - 100% of privileged operations protected by RBAC.
- **Validation Method:**
    - Automated and manual role/permission tests.
- **Failure Conditions:**
    - Privilege escalation, unauthorized actions.
- **Acceptance Threshold:**
    - No privilege escalation possible.

3.3 Data Protection

- **Acceptance Criteria:**
    - All sensitive data at rest must be encrypted (AES-256 or equivalent).
    - All data in transit must use HTTPS/TLS 1.2+.
    - API secrets/tokens never logged or exposed.
- **Test Scenarios:**
    - Penetration testing, log review.
- **Success Metrics:**
    - 100% compliance with encryption policies.
- **Validation Method:**
    - Security audits, automated tools (OWASP ZAP).
- **Failure Conditions:**
    - Unencrypted data, credentials in logs.

- **Acceptance Threshold:**
    - Zero critical vulnerabilities.

3.4 Input Validation & Security Headers

- **Acceptance Criteria:**
    - All inputs validated/sanitized against schema; reject malicious data (SQLi, XSS, etc.).
    - HTTP security headers (e.g., Content-Security-Policy, X-Content-Type-Options, X-Frame-Options, Strict-Transport-Security) set on all responses.
- **Test Scenarios:**
    - Fuzz and injection attacks; check response headers.
- **Success Metrics:**
    - No successful injection attacks; all required headers present.
- **Validation Method:**
    - Automated security scanning.
- **Failure Conditions:**
    - Vulnerable endpoints, missing headers.
- **Acceptance Threshold:**
    - 100% endpoints pass input validation and header checks.

---

4. Reliability and Availability Criteria

4.1 Uptime Requirements

- **Acceptance Criteria:**
    - System must maintain ≥99.5% uptime excluding scheduled maintenance.
    - Maintenance windows must be configurable and communicated.
- **Test Scenarios:**
    - Track uptime via health checks and logs over 30 days.
- **Success Metrics:**
    - ≥99.5% uptime recorded.
- **Validation Method:**
    - Monitoring tools, uptime reports.
- **Failure Conditions:**
    - Unscheduled outages, missed downtime notifications.
- **Acceptance Threshold:**
    - Meets SLA over rolling 30-day window.

4.2 Error Handling, Fault Tolerance & Recovery

- **Acceptance Criteria:**
  - All critical failures logged with actionable details.
  - System recovers automatically from provider/API failures (with retry/fallback).
  - Data backups run daily; restore tested quarterly.
- **Test Scenarios:**
  - Simulate component/API failures, data corruption, forced restore from backup.
- **Success Metrics:**
  - 100% recovery from test failures; zero data loss in backup/restore tests.
- **Validation Method:**
  - Automated failover and recovery tests; manual backup/restore drills.
- **Failure Conditions:**
  - Data loss, unrecoverable errors.
- **Acceptance Threshold:**
  - 100% recovery in test scenarios.

---

5. Compatibility and Integration Criteria

5.1 Browser and Platform Compatibility

- **Acceptance Criteria:**
  - Admin UI supports latest two versions of Chrome, Firefox, Edge, Safari.
  - CLI supports Windows, macOS, Linux (Node.js ≥18).
- **Test Scenarios:**
  - UI smoke tests on all platforms; CLI functional tests on supported OSes.
- **Success Metrics:**
  - 100% of core features work on all platforms.
- **Validation Method:**
  - Manual/automated cross-platform tests.
- **Failure Conditions:**
  - Feature failures, major UI bugs on any platform.
- **Acceptance Threshold:**
  - No critical issues on any supported platform.

5.2 API and Third-party Integration Compatibility

- **Acceptance Criteria:**

    - API endpoints must maintain backward compatibility for at least one major version.
    - Integration modules (Adobe, Confluence, SharePoint) must adapt to minor upstream API changes.
- **Test Scenarios:**
    - Run integration suite against current and previous API versions.
    - Simulate breaking changes in third-party APIs.
- **Success Metrics:**
    - No breaking changes without major version increment; integrations adapt to minor changes.
- **Validation Method:**
    - Automated regression and integration tests.
- **Failure Conditions:**
    - Breaking API changes, integration failures.
- **Acceptance Threshold:**
    - 100% backward compatibility for at least one version.

5.3 Legacy System Integration

- **Acceptance Criteria:**
    - Data import/export to legacy systems must function per defined mappings.
    - Document migration scripts provided and tested.
- **Test Scenarios:**
    - Import/export tests with legacy datasets.
- **Success Metrics:**
    - 100% data mapping accuracy.
- **Validation Method:**
    - Manual/automated data migration tests.
- **Failure Conditions:**
    - Data mapping errors, migration failures.
- **Acceptance Threshold:**
    - 0% critical migration failures.

---

6. Quality and Maintainability Criteria

6.1 Code Quality

- **Acceptance Criteria:**
    - Code coverage $\geq$90% (unit + integration).
    - Code must pass ESLint (Airbnb) and Prettier formatting.
    - Cyclomatic complexity $<$10 for all functions.

- **Test Scenarios:**
    - Run code coverage, lint, and complexity analysis.
- **Success Metrics:**
    - ≥90% coverage, 0 lint errors, all complexity checks pass.
- **Validation Method:**
    - CI pipeline (Jest, ESLint, complexity tools).
- **Failure Conditions:**
    - Low coverage, linting failures, high complexity.
- **Acceptance Threshold:**
    - All code quality gates passed on CI.

6.2 Documentation

- **Acceptance Criteria:**
    - All public APIs, modules, and configuration options fully documented.
    - User/developer documentation updated for every major/minor release.
- **Test Scenarios:**
    - Review and test documentation for completeness and accuracy.
- **Success Metrics:**
    - No undocumented public APIs; docs ≥95% accurate in user testing.
- **Validation Method:**
    - Documentation review, user onboarding tests.
- **Failure Conditions:**
    - Outdated, incomplete, or missing docs.
- **Acceptance Threshold:**
    - 100% of features documented before release.

6.3 Testing Requirements

- **Acceptance Criteria:**
    - Automated tests for unit (≥90%), integration (≥80%), and end-to-end (≥80%) scenarios.
    - All critical bugs must have associated regression tests.
- **Test Scenarios:**
    - Run all test suites, inject regression scenarios.
- **Success Metrics:**
    - All suites pass; no test regressions.
- **Validation Method:**

- CI/CD pipeline, manual regression verification.

- **Failure Conditions:**
    - Broken builds, failing regression tests.
- **Acceptance Threshold:**
    - 0 critical test failures.

6.4 Monitoring and Logging

- **Acceptance Criteria:**
    - System emits structured logs for all API and integration events (Winston, express-winston).
    - Health and performance metrics exposed via Prometheus endpoints.
- **Test Scenarios:**
    - Trigger API/integration events; check logs and metrics endpoints.
- **Success Metrics:**
    - 100% of critical events logged; metrics available.
- **Validation Method:**
    - Manual/automated log and metric inspection.
- **Failure Conditions:**
    - Missing logs, silent failures.
- **Acceptance Threshold:**
    - All key events observable.

6.5 Configuration Management

- **Acceptance Criteria:**
    - All environment and provider configs must be externalized (dotenv, JSON, or environment variables).
    - CLI and API must support dynamic reloading without restart (where feasible).
- **Test Scenarios:**
    - Change config at runtime; verify effect.
- **Success Metrics:**
    - Config changes applied without downtime (where supported).
- **Validation Method:**
    - Manual/automated config reload tests.
- **Failure Conditions:**
    - Hard-coded configs, config reload failures.
- **Acceptance Threshold:**
    - 100% external config; dynamic reloads work as documented.

7. Validation Methods and Test Scenarios (Summary Table)

Criterion Area	Validation Method	Test Scenarios	Success Metric	Failure Condition	Accepta
API Functionality	Automated API tests, Swagger UI	Valid/invalid payloads, error paths	100% endpoint test pass	Undocumented, unvalidated endpoints	100% pa errors
Data Processing	Unit/integration tests, fuzzing	Valid/invalid data, concurrency	0% corruption, 100% schema pass	Data loss, schema fail	99.99% i
Business Logic	Unit tests, diffing, failover	Output vs reference, provider failover	100% rule/test coverage	Calculation/rule failures	100% rul
Integration Points	E2E integration, manual	Auth, data exchange, error handling	99% reliability	Auth/data/protocol failures	99% pas
User Interface	UI tests, Lighthouse/manual	Browser/device, accessibility	≥90 accessibility score	Broken UI, inaccessible elements	No critic
Performance	Load/soak tests, monitoring	Load, concurrent jobs, scaling	95% requests within SLA	Degraded/slow response	95% SLA
Security	Automated/manual security tests	AuthZ/AuthN, input validation, headers	100% protection	Unauth access, vulnerabilities	100% pa issues
Reliability	Monitoring, backup drills	Uptime, failure/recovery, backup/restore	≥99.5% uptime, 100% recovery	Downtime, unrecoverable errors	Meets SL recovery
Compatibility	Cross-platform, regression tests	All platforms, API versions, legacy integration	100% features on all platforms	Platform-specific failures	0 critical
Code Quality	Coverage, lint, complexity tools	All modules/functions	≥90% coverage, 0 lint errors	Low coverage, linting failures	All qualit
Documentation	Review, user tests	All APIs, modules, configs	95% user accuracy	Missing/outdated docs	100% fea documen

Criterion Area	Validation Method	Test Scenarios	Success Metric	Failure Condition	Accepta
Monitoring/Logging	Log/metrics review	API/integration events, metrics	100% key events observable	Silent failures	All key e observab
Config Management	Manual/automated reload tests	Runtime config changes	100% external config, reloadable	Hardcoded config, reload failures	100% externali

---

All acceptance criteria must be met for release to production. Deviations require explicit approval, with documented risk and mitigation.

End of Technical Acceptance Criteria