Test Plan

Source File: generated-documents\quality-assurance\test-plan.md
Generated: 16/07/2025 at 13:58:55
Generated by: Requirements Gathering Agent - PDF Converter

Test Plan

Generated by adpa-enterprise-framework-automation v3.2.0
Category: quality-assurance
Generated: 2025-07-14T21:00:40.072Z
Description: Detailed test plan with test scenarios and execution plan

```
# Test Plan
**Project:** ADPA - Advanced Document Processing & Automation Framewor
**Version:** 3.2.0
**Date:** July 2025
**Owner:** QA/Test Manager
**Document Status:** Approved for Execution

---

## 1. Test Plan Overview

### 1.1 Purpose
This test plan defines the testing scope, approach, resources, schedul

### 1.2 Scope & Objectives
- **In Scope:**
  - Functional, integration, security, performance, and usability test
  - Verification of compliance with BABOK v3, PMBOK 7th Ed., and DMBOK
  - End-to-end testing of document generation pipelines and enterprise

- **Out of Scope:**
  - DMBOK 2.0 modules not yet implemented.
  - Mobile application support (future roadmap).

- **Objectives:**
  - Validate all core features against requirements and standards.
  - Ensure system is production-ready, secure, and performant.
  - Guarantee integrations (Adobe, Confluence, SharePoint, AI provider
  - Achieve defined quality and compliance criteria prior to release.

### 1.3 Project Background & Context
ADPA is a modular, standards-compliant Node.js/TypeScript enterprise f

### 1.4 Assumptions & Constraints
- All test environments mirror production configurations.
- Test data is anonymized and compliant with data protection regulatio
- Third-party API keys and credentials are available for testing.
- Some features (e.g., DMBOK 2.0) are partially implemented and will b
- Enterprise integrations (Adobe, Confluence, SharePoint) test account

---

## 2. Test Items and Features

### 2.1 Features/Modules to be Tested
| Module                           | Features/Endpoints
|----------------------------------|----------------------------------
| **Core Engine**                  | AI orchestration, context managem
| **CLI Interface**                | Command parsing, batch generation
| **REST API**                     | `/api/v1/generate`, `/api/v1/temp
| **Admin Web Interface**          | User authentication, document man
| **AI Provider Integrations**     | OpenAI, Google AI, Copilot, Ollam
| **Adobe Integration**            | InDesign, Illustrator, Photoshop,
| **Confluence Integration**       | Authentication, publishing, statu
| **SharePoint Integration**       | Authentication, upload, folder ma
| **Compliance Modules**           | BABOK v3, PMBOK 7th, DMBOK 2.0 (w
| **Security**                     | Authentication, authorization, ra
| **Performance/Scalability**      | Load handling, failover, provider
| **Usability**                    | CLI, API, Web UI experience, erro

### 2.2 Version Identification & Build Info
```

```
- **Software Version:** 3.2.0
- **API Specification:** OpenAPI 3.0, TypeSpec
- **Build Artifacts:** NPM package, Docker image (pending), Next.js Ad

### 2.3 Dependencies & Integration Points
- External APIs: OpenAI, Google AI, GitHub Copilot, Ollama, Adobe, Atl
- Node.js 18+, TypeScript 5.7+, Express.js
- OAuth2, API keys, JWT for authentication
- Azure/Microsoft Graph for SharePoint
- Active Directory/SAML (planned)
- Database: JSON config, extensible to SQL/NoSQL

---

## 3. Test Approach and Strategy

### 3.1 Testing Levels
- **Unit Testing:** Core logic, utility functions, template processors
- **Integration Testing:** API endpoints, CLI commands, module interco
- **System Testing:** End-to-end workflows (document generation, publi
- **User Acceptance Testing (UAT):** Business scenarios, enterprise us

### 3.2 Testing Types
- **Functional:** Requirement coverage, business rules, standards comp
- **Performance:** API throughput, response times, concurrent user han
- **Security:** Auth flows, permission checks, OWASP vulnerabilities,
- **Usability:** CLI experience, web UI navigation, API documentation
- **Compatibility:** Node.js/TypeScript versions, browser compatibilit
- **Regression:** Automated on each build/release.
- **Integration:** Third-party APIs (Adobe, Confluence, SharePoint), p

### 3.3 Test Design Techniques & Methodologies
- **Requirement-based Testing:** Direct mapping to user stories and st
- **Boundary Value & Equivalence Partitioning:** Input validation, API
- **Exploratory Testing:** Ad-hoc scenarios for integrations and edge
- **Data-driven Testing:** Multiple test datasets for document generat
- **Risk-based Testing:** Prioritize critical business and compliance

### 3.4 Automation Strategy & Tools
- **Unit/Integration Tests:** Jest, ts-jest for automated execution, c
- **API Automation:** Postman/Newman, supertest for REST API endpoints
- **CLI Automation:** Shell scripts, expect, and custom Node.js test h
- **Web UI Automation:** Playwright or Cypress for Admin portal.
- **Continuous Integration:** GitHub Actions for automated builds and
- **Performance Testing:** Artillery or k6 for load and stress tests.

---

## 4. Test Environment Requirements

### 4.1 Hardware/Software
- **Hardware:**
  - x64 architecture, minimum 8GB RAM, SSD storage
  - Azure/AWS VM or local equivalent for parallel test environments

- **Software:**
  - Node.js 18+
  - TypeScript 5.7+
  - NPM/Yarn
  - Docker (where available)
  - Browsers: Chrome, Edge, Firefox (for admin portal)

### 4.2 Test Data Management
- **Test Data:**
  - Anonymized project/sample data for document generation
  - Predefined templates for BABOK, PMBOK, DMBOK modules
  - Test user accounts for all roles (admin, analyst, manager, viewer)
  - API credentials for all external providers

- **Management:**
  - Versioned test data sets in `/test/data`
  - Data reset scripts for environment teardown/setup

### 4.3 Environment Setup & Configuration
- **Setup Script:**
  - `npm run dev` (development), `npm run api:start`, `npm run admin:s
  - `.env` configuration for API keys and endpoints
  - OAuth2 configuration for Adobe, Confluence, SharePoint
- **Configuration Procedures:**
  - Step-by-step documented in `/docs/` and onboarding guides
  - Automated setup scripts for repeatability

### 4.4 Access & Security
- **Access:**
  - Controlled via test user accounts with role-based permissions
  - Secure storage of secrets and credentials (not in source code)
- **Security:**
  - SSL/TLS for API endpoints (where possible)
  - Encrypted test data where sensitive

---

## 5. Test Schedule and Milestones

### 5.1 Timeline & Phases

| Phase                 | Start Date | End Date   | Deliverable
|-----------------------|------------|------------|--------------------
| Test Planning         | 2025-07-10 | 2025-07-12 | Approved Test Plan
| Environment Setup     | 2025-07-13 | 2025-07-14 | Configured Test Env
| Unit Testing          | 2025-07-15 | 2025-07-18 | Unit Test Report
| Integration Testing   | 2025-07-19 | 2025-07-22 | Integration Test Re
| System Testing        | 2025-07-23 | 2025-07-27 | System Test Report
| UAT & Compliance      | 2025-07-28 | 2025-07-30 | UAT Summary, Compli
| Performance/Security  | 2025-07-28 | 2025-07-30 | Perf/Security Test
| Regression Testing    | 2025-07-28 | 2025-07-30 | Regression Test Rep
| Test Closure          | 2025-07-31 | 2025-08-01 | Test Completion Rep

### 5.2 Key Milestones
- Test Plan Approval
- Environment Ready
- All Unit Tests Pass
- All Integration Tests Pass
- System Test Exit
- UAT Completion
- Test Sign-off

### 5.3 Dependencies & Critical Path
- External API availability (Adobe, Confluence, SharePoint)
- Credential provisioning for test accounts
- Timely delivery of new/updated modules (DMBOK 2.0, Adobe Phase 2)
- CI/CD pipeline stability

### 5.4 Resource Allocation
- Test Lead (100%)
- 2 QA Engineers (80%)
- 1 Automation Engineer (50%)
- 1 DevOps Support (20%)
- 2 Business SMEs (UAT, 20% during UAT window)
```

```
---

## 6. Test Team Organization

### 6.1 Roles & Responsibilities

| Role               | Responsibilities
|-------------------|-------------------------------------------------
| Test Manager       | Overall QA planning, reporting, escalation
| QA Engineer        | Test case design, manual/automated execution, de
| Automation Eng.    | Maintain automation scripts, CI integration
| DevOps Support     | Environment setup, build/release support
| Business SME       | UAT, standards compliance validation
| Developer          | Defect resolution, support root cause analysis

### 6.2 Skills & Competencies
- Strong experience with Node.js/TypeScript development & testing
- Familiarity with enterprise integration (OAuth2, REST APIs)
- Knowledge of BABOK, PMBOK, DMBOK frameworks
- Automation expertise (Jest, Cypress/Playwright, k6)
- Security and compliance awareness

### 6.3 Communication & Reporting
- Daily standups during execution
- Weekly status reports to stakeholders
- Defect triage meetings (as needed)
- Test dashboards (via CI tools)

### 6.4 Escalation Procedures
- Critical/blocker issues escalated to Test Manager within 2 hours
- Test Manager escalates to Project Manager/Dev Lead as required
- SLA for resolution based on impact (see risk management)

---

## 7. Entry and Exit Criteria

### 7.1 Entry Criteria (by phase)
- **Unit Testing:** Code committed, dev environment ready, unit tests
- **Integration Testing:** All unit tests pass, integration environmen
- **System Testing:** Core modules integrated, basic smoke tests pass,
- **UAT:** All system tests passed, UAT environment configured, user g
```

```
### 7.2 Exit Criteria / Definition of Done
- All planned test cases executed (≥ 95% coverage)
- All critical and high-priority defects fixed and retested
- No open critical/blocker defects
- Regression suite passes
- UAT sign-off received
- Test documentation delivered

### 7.3 Suspension & Resumption Criteria
- Testing suspended for:
  - Critical environment outages
  - Blocking external dependencies (e.g., unavailable API)
- Resumed when:
  - Blocking issues resolved, verified by Test Manager

### 7.4 Risk-based Decision Points
- If unresolved high-severity defects are not business-critical, proce

---

## 8. Test Deliverables

- **Test Plan** (this document)
- **Test Cases & Scripts** (manual and automated, version controlled)
- **Test Data Sets** (anonymized, versioned)
- **Test Execution Reports** (per phase, summarized)
- **Defect Reports** (JIRA/GitHub, with root cause and status)
- **Performance & Security Test Reports**
- **Traceability Matrix** (requirements to test cases)
- **Test Completion Report** (including coverage, quality metrics, les

---

## 9. Risk Management

### 9.1 Identified Risks & Impact

| Risk                                             | Impact
|--------------------------------------------------|------------------
| External API downtime (Adobe, AI, etc.)          | Test delays, cove
| Incomplete third-party integration (DMBOK 2.0)   | Partial coverage
| Test data privacy/compliance issues              | Regulatory, block
| Environment configuration drift                   | Inconsistent resu
| Credential leaks or mismanagement                 | Security, complia
| Performance issues under load                     | Production risk
| Resource constraints/availability                 | Schedule delays

### 9.2 Mitigation & Contingency
- Maintain stubs/mocks for critical third-party dependencies.
- Establish backup test environments.
- Frequent environment validation and configuration-as-code.
- Reserve test slots for retesting after critical fixes.

### 9.3 Risk Monitoring & Escalation
- Weekly risk review in QA status meeting.
- Immediate escalation for blocker/critical risks.
- Document and track in project risk register.

---

## 10. Approval and Sign-off

### 10.1 Review & Approval Process
- Review by QA Lead, Project Manager, and Lead Developer.
- Stakeholder review for UAT and compliance sign-off.

### 10.2 Stakeholder Sign-off
- Test Plan approved by QA Lead and Project Manager.
- Test Completion Report signed by QA Lead, Project Manager, and Produ

### 10.3 Change Management
- All changes to test scope, approach, or major deliverables must be r
- Version control maintained for all test documentation.
- Change log maintained and communicated to all stakeholders.

---

**Prepared by:**
[QA/Test Manager Name]
**Date:** [Insert Date]

**Approved by:**
[Project Manager Name]
[Product Owner Name]
[Lead Developer Name]
```

```
**Date:** [Insert Date]

---

**End of Test Plan Document**
```