Quality Management Plsn

> **Source File:** generated-documents\planning\quality-management-plsn.md
> **Generated:** 30/07/2025 at 06:59:07
> **Generated by:** Requirements Gathering Agent - PDF Converter

QualityManagementPlsn

Generated by adpa-enterprise-framework-automation v3.2.0
Category: planning
Generated: 2025-07-14T21:31:44.189Z
Description:

Quality Management Plan

Project: ADPA – Advanced Document Processing & Automation Framework
Version: 3.2.0
Author: ADPA Core Team
Date: July 2025
License: MIT
Repository: GitHub
Documentation: Wiki

---

1. Purpose

This Quality Management Plan (QMP) defines the quality objectives, assurance activities, control measures, and continuous improvement processes for the ADPA Enterprise Framework. ADPA is a modular, standards-compliant Node.js/TypeScript automation framework for enterprise requirements, project, and data management, supporting CLI and REST API interfaces, and integrating with enterprise platforms such as Adobe, Confluence, and SharePoint.

---

2. Quality Objectives

- **Standards Compliance:** Automate and generate documentation aligned with BABOK v3, PMBOK 7th Edition, and DMBOK 2.0.
- **Reliability:** Ensure robust functionality across all modules (AI provider orchestration, document generation, integration layers).
- **Security:** Maintain enterprise-grade security, including authentication, authorization, and regulatory compliance (GDPR, SOX, PCI DSS).
- **Usability:** Deliver a seamless user experience via CLI, REST API, and web admin portal.
- **Interoperability:** Guarantee integration with enterprise systems (Confluence, SharePoint, Adobe, VCS, Identity Management).
- **Performance & Scalability:** Support large-scale, multi-user deployments with high performance and horizontal scalability.
- **Maintainability:** Enforce clean code standards, modular design, and comprehensive documentation.

---

3. Quality Assurance Approach

3.1 Standards & Frameworks

- **BABOK v3, PMBOK 7, DMBOK 2.0:** All generated outputs and automation workflows are validated against these standards.
- **ISO 9001 & 27001:** Internal processes align with these standards for quality and information security.

- **Industry Compliance:** Basel III, MiFID II, FINRA, CFTC, FCA, BaFin, SOX, PCI DSS, HIPAA, FedRAMP.

3.2 Development Practices

- **TypeScript Strict Mode:** Enforced in all modules.
- **Automated Code Linting:** Airbnb ESLint configuration, Prettier formatting.
- **Unit & Integration Testing:** Jest-based test suites with >90% coverage targets.
- **Conventional Commits:** For traceable and standardized commit history.

3.3 Documentation

- **Comprehensive API Docs:** OpenAPI specs generated via TypeSpec, published with Swagger and Redoc.
- **Framework Templates:** All document templates are version-controlled and peer-reviewed.
- **User Guides:** Maintained for CLI, API, and all integrations; kept up-to-date with releases.

---

4. Quality Control Activities

4.1 Code Review

- **Mandatory Peer Review:** All code changes are reviewed via pull requests.
- **Automated CI Checks:** Lint, build, and test pipelines must pass before merge.

4.2 Testing Strategy

- **Unit Testing:** Each core component and utility function is covered.

- **Integration Testing:** Focus on end-to-end workflows (document generation, publishing, API endpoints).
- **Provider-Specific Testing:** Dedicated scripts (e.g., `test:azure`, `test:github`, `test:ollama`) for AI provider integrations.
- **Performance Testing:** Regular execution of `npm run test:performance` to validate system scalability.
- **Security Testing:** Penetration tests and static analysis for vulnerabilities.
- **Regression Testing:** Automated on each release candidate build.

4.3 Release Management

- **Version Tagging:** Semantic versioning with a clear changelog.
- **Release Checklists:** Each release must pass predefined acceptance criteria (test coverage, documentation, deployment verification).
- **Rollback Procedures:** Documented and tested for all deployment environments.

---

5. Quality Metrics & Monitoring

- **Test Coverage:** Target >90% for critical modules.
- **API Response SLAs:** 99.9% uptime for REST API endpoints in production.
- **Defect Density:** <1 major defect per release.
- **User Feedback:** Collected via GitHub Issues and Community Discussions; tracked and trended.
- **Performance Benchmarks:** Document generation time, API response latency, throughput under load.
- **Compliance Audit Logs:** All document outputs and API requests are logged for auditability.

---

6. Roles & Responsibilities

Role	Responsibilities
Product Owner	Approves quality objectives and release criteria
QA Lead	Defines and enforces quality processes, oversees all testing and review efforts
Developers	Implement features, write unit/integration tests, and maintain code quality
DevOps Engineer	Maintains CI/CD pipelines, monitors system health, ensures deployment quality
Documentation Lead	Maintains all user, API, and system documentation
Security Officer	Conducts security reviews, manages compliance audits

7. Tools & Infrastructure

- **CI/CD:** GitHub Actions for automated build, test, and deployment.
- **Linting & Formatting:** ESLint (Airbnb), Prettier.
- **Testing:** Jest, ts-jest, custom provider test scripts.
- **Documentation:** TypeSpec, Swagger UI, Redocly, Markdown Wiki.
- **Monitoring:** Built-in metrics endpoints, health checks, and external observability tools (e.g., Azure Monitor).
- **Code Repository:** GitHub with protected branches and required review status checks.
- **Issue Tracking:** GitHub Issues, project boards for release management.

8. Integration Quality Management

- **Adobe Integration:** Follow Adobe SDK authentication and output validation guides. Regularly update credentials and revalidate template outputs.
- **Confluence/SharePoint:** OAuth2 authentication, publish flow tested with both sandbox and production tenants; metadata and versioning verified.
- **AI Providers:** Automated failover tested; provider response consistency and accuracy validated with golden datasets.
- **API:** OpenAPI schema validation, endpoint contract testing, and backward compatibility checks per each release.

---

9. Continuous Improvement

- **Retrospectives:** After each release, the team reviews quality outcomes and updates this plan.
- **Root Cause Analysis:** All critical defects require postmortem analysis and process improvement proposals.
- **User Feedback Loops:** Regular review of issue tracker and discussion forums for quality improvement insights.
- **Training:** Ongoing developer and QA training in standards, security, and testing best practices.

---

10. Review & Approval

This Quality Management Plan is reviewed quarterly and upon any major architectural or process change. Updates are approved by the Product Owner and QA Lead.

Appendix: References

- BABOK v3 Standard
- PMBOK 7th Edition
- DMBOK 2.0
- ISO 9001:2015
- ISO 27001
- ADPA Documentation

Quality is not an act, it is a habit. — ADPA Team

Generated from generated-documents\planning\quality-management-plsn.md | Requirements Gathering Agent